



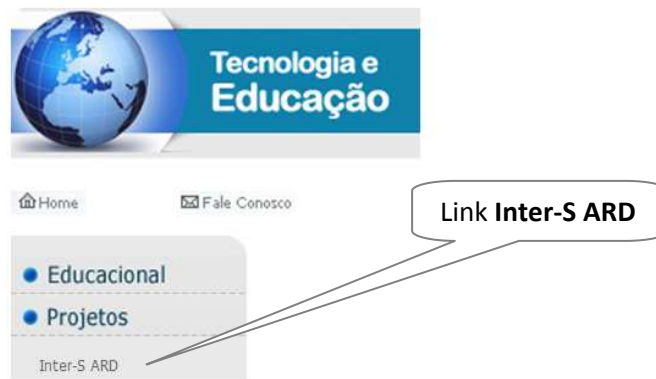
Projetos INTER-S com Arduino

Sumário

1. Instalação do Inter-S.....	2
2. Instalação do software Arduino.....	3
3. Instruções INTER-S para comunicação com Arduino.....	8
4. Projetos.....	10
4.1. Projeto 1: Acionamento de um LED.....	10
4.2. Projeto 2: Sequência de 8 LEDs.....	15
4.3. Projeto 3: Display de 7 segmentos.....	18
4.4. Projeto 4: Controle de um servomotor.....	22
4.5. Projeto 5: Braço robótico (controle de 4 servos).....	26
4.6. Projeto 6: Acionamento de um LED RGB (saída analógica).....	34
4.7. Projeto 7: Reconhecimento de botões (entradas digitais).....	39

1. Instalação do Inter-S

Acesse o site <http://vichinsky.com.br> e localize o item Projetos no menu esquerdo. Clique sobre o link Inter-S ARD.



Faça o download da versão ARD 2012 e execute o programa de instalação.

Na mesma página de download, baixe o arquivo INTER_S_1.ino. Este arquivo contém o programa que deverá ser “embarcado” na placa de prototipagem (os procedimentos para o *upload* deste programa estão descritos na seção 2)

ATENÇÃO usuários do Windows VISTA e Windows 7 (seven)

Os sistemas Windows VISTA e Seven têm um controle de conta do usuário que pode bloquear o registro de alguns componentes do INTER-S (dll's e OCX's), tornando-os assim inoperantes. Para contornar este problema, antes de instalar o INTER-S você deve desabilitar o serviço de Controle de Conta de Usuário (UAC - *User Account Control*). Veja abaixo como desabilitar o UAC:

- Acesse o menu **INICIAR** do Windows e selecione a opção **Painel de Controle**
- Acesse a opção **Contas de Usuário e Segurança Familiar**
- Selecione o item **Contas de Usuário**
- Se seu sistema for o VISTA, clique na opção **Ativar ou desativar o Controle de Conta de Usuário** e desabilite a opção "**Utilizar o Controle de Conta de Usuário**";
- Se seu sistema for o Seven, clique na opção **Alterar configurações de Controle de Conta de Usuário** e desloque a barra deslizante para o nível "**Nunca notificar**";
- Confirme a operação clicando sobre o botão **OK**;
- Reinicie o computador para que a configuração seja efetuada.

Após o download do pacote de instalação, execute-o e siga as orientações de tela. Aceite todas as opções padrões.

Usuários do Windows VISTA ou SEVEN (7) devem reativar o Controle de Conta de Usuário (siga os procedimentos descritos acima para reativá-lo).

2. Instalação do software Arduino

Acesse o site oficial <http://arduino.cc> e localize o item Download no menu superior. Surgirá uma nova página onde você deverá baixar o aplicativo adequado para o seu sistema operacional.

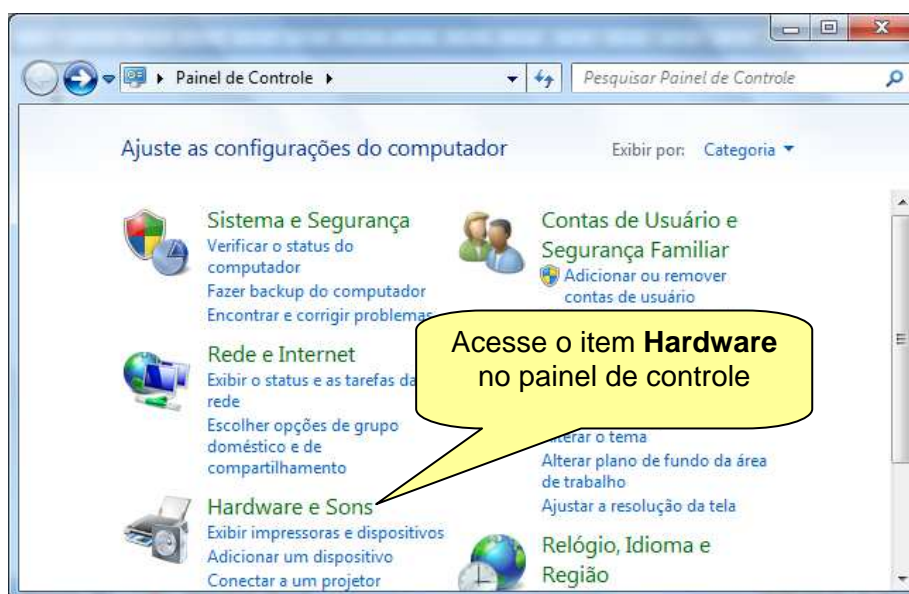


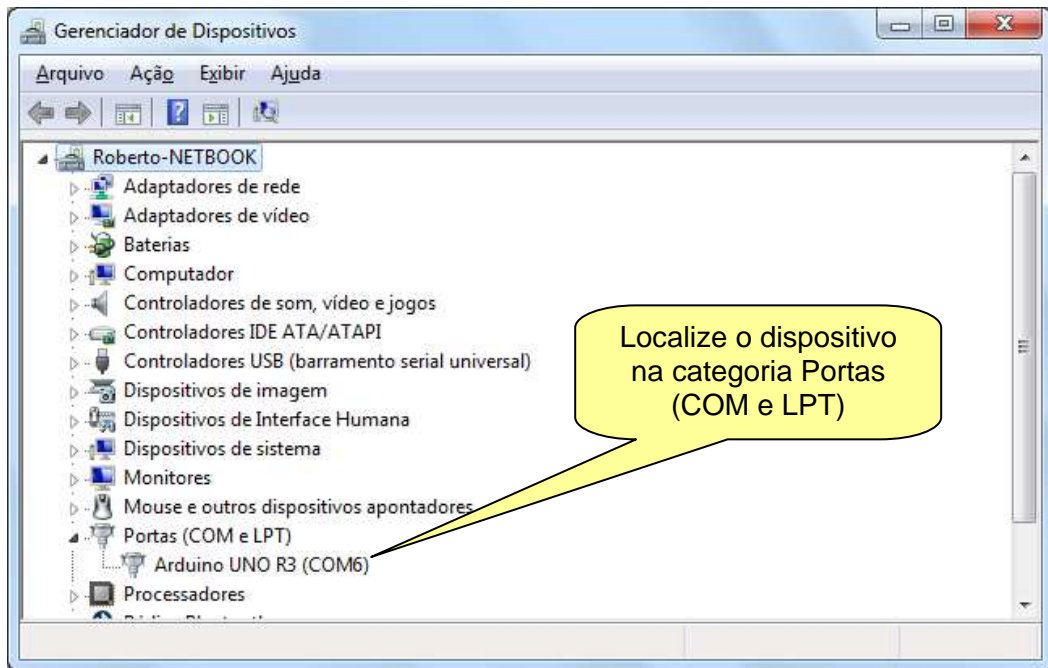
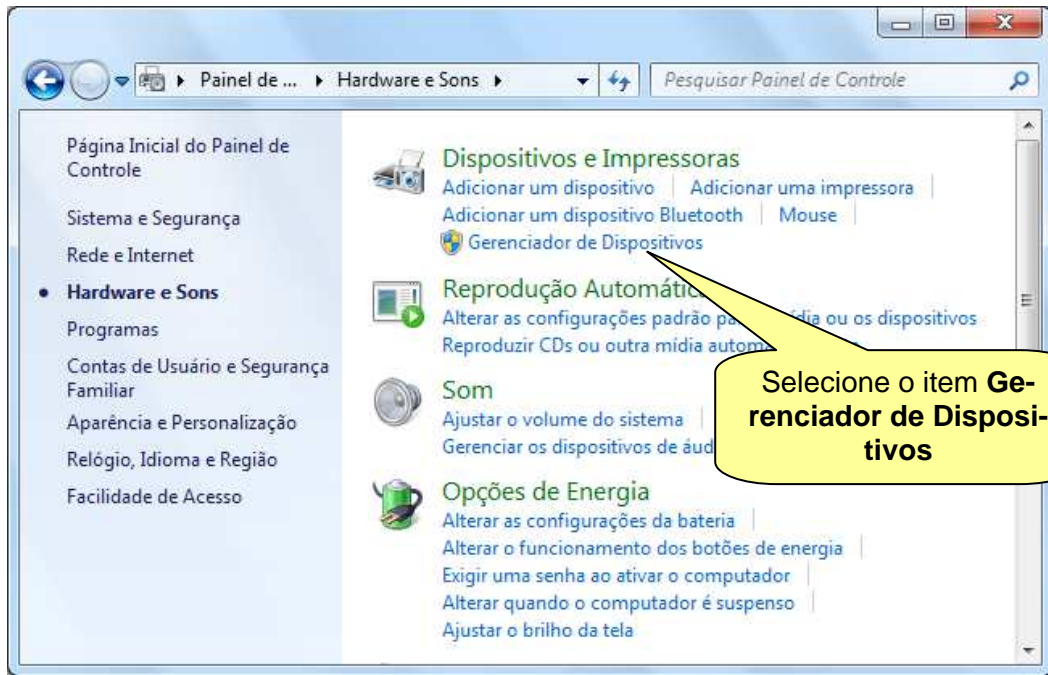
O software Arduino vem em um arquivo compactado e não necessita de instalação, basta descompactá-lo em um local de sua preferência. É altamente recomendável que você crie um atalho no *desktop* para o programa arduino.exe que se encontra na pasta descompactada.

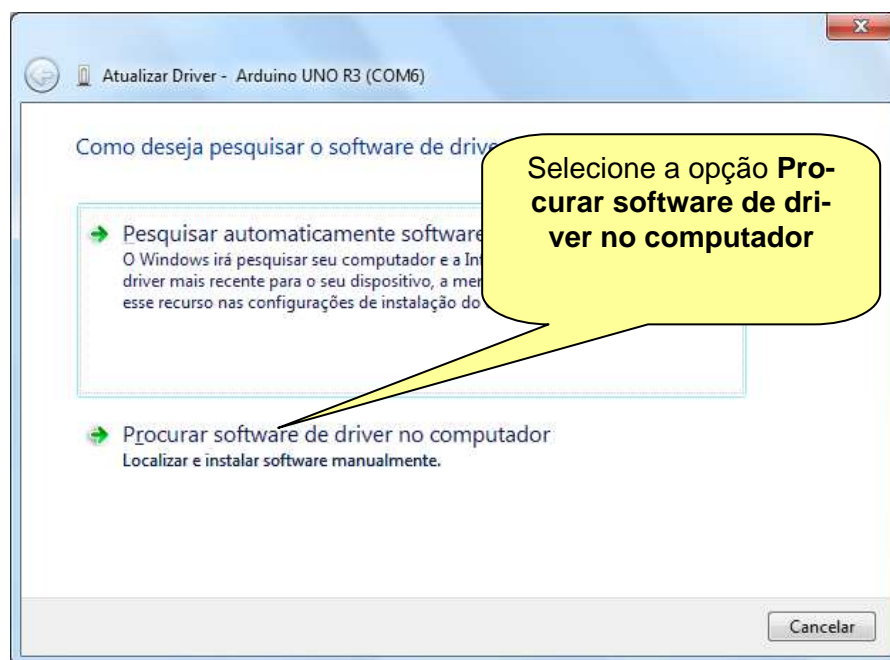
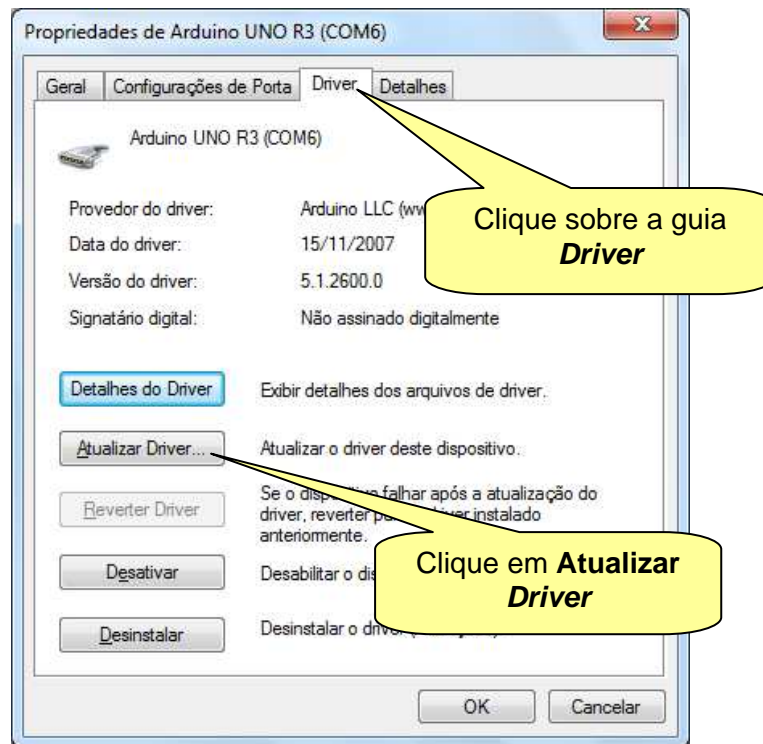
Ainda será necessário instalar o *driver* da placa de prototipagem. Normalmente este *driver* é instalado automaticamente quando a placa é conectada no computador pela primeira vez.

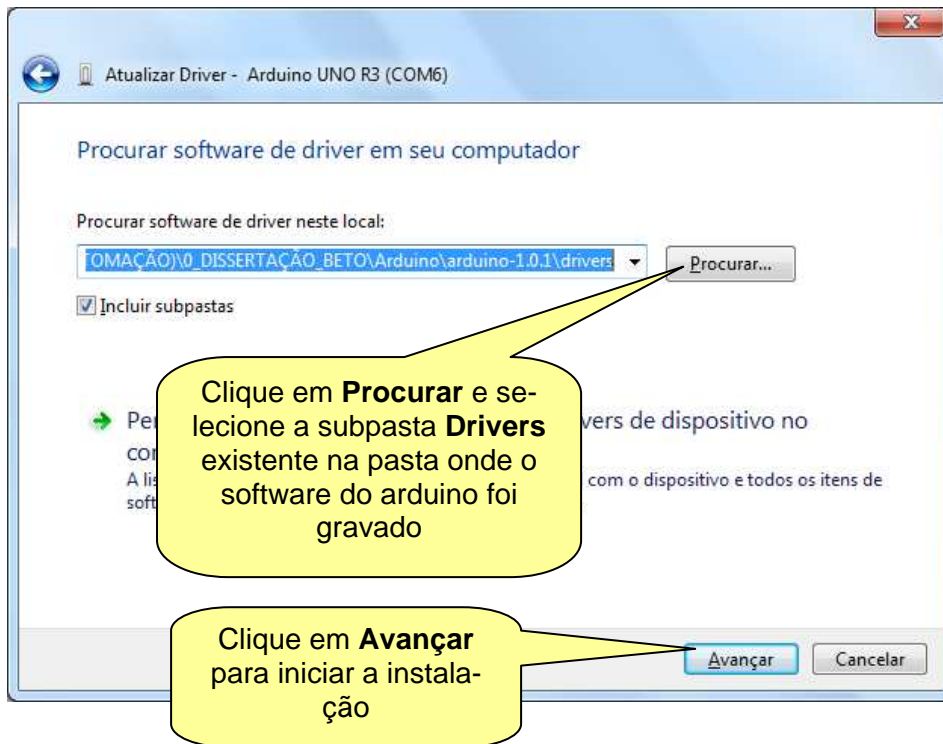
Porém, existem ocasiões em que o *driver* não é corretamente instalado. Nestes casos, você deve promover a instalação manualmente através do painel de controle, encontrando o dispositivo e atualizando o *driver* com aqueles que se encontram na subpasta **drivers** da pasta do software do arduino.

Procedimentos para instalação manual do *driver* (se necessário)





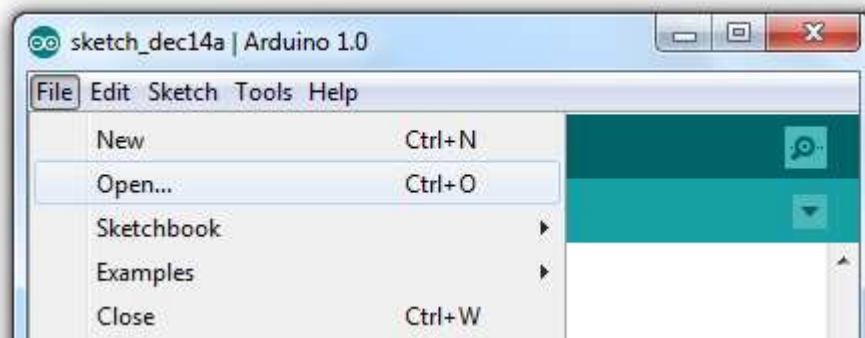




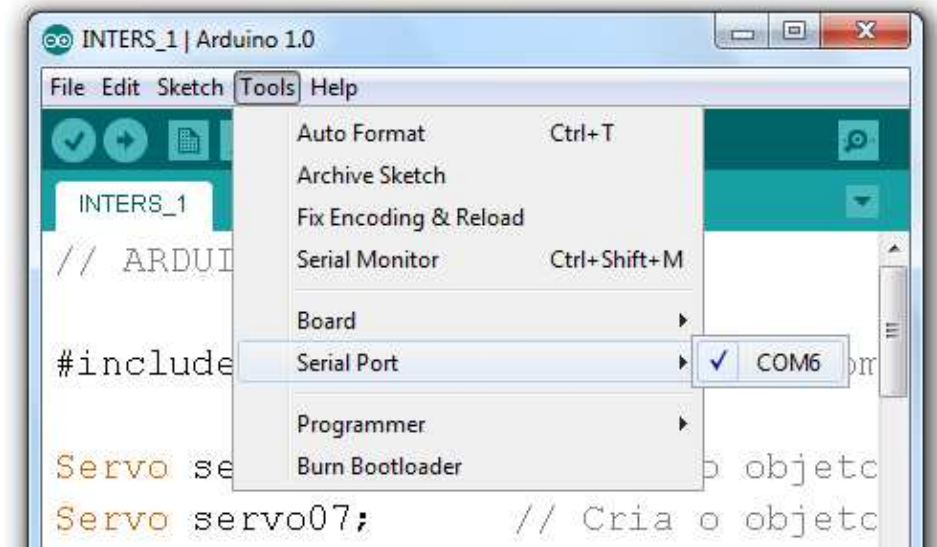
Procedimentos para “embarcar” o programa na placa de prototipagem Arduino

O programa INTERS_1.ino, cujo download deve ser efetuado através do site <http://vichinsky.com.br> (conforme orientações dadas na seção 1), deve ser gravado (embarcado) na placa Arduino. Para esta tarefa, siga os procedimentos descritos a seguir:

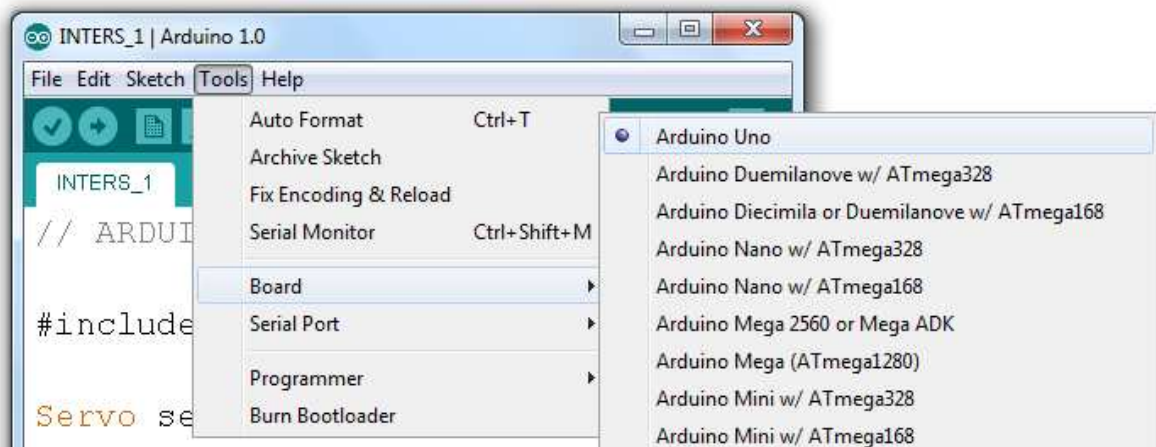
- a. Conecte a placa de prototipagem no seu computador;
- b. Execute o programa **arduino.exe** que se encontra na pasta onde o software Arduino foi gravado (recomendamos que você crie um atalho na área de trabalho para que o programa arduino.exe possa ser encontrado com mais facilidade futuramente);
- c. Carregue o programa INTERS_1.ino através do comando **Open** do menu **File**:



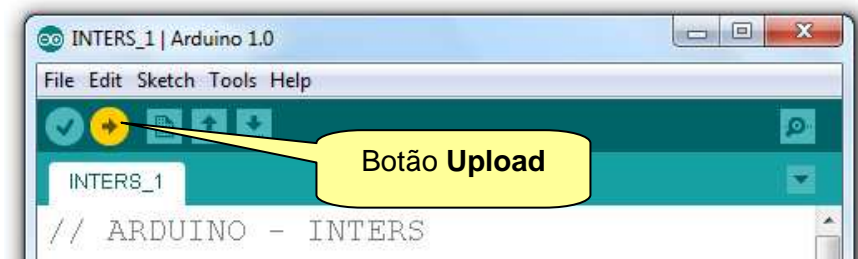
- d. Na opção **Serial Port** do menu **Tools**, verifique se a porta COM foi marcada corretamente para a placa conectada. Caso não apareça nenhuma opção de porta, verifique se a ligação do cabo está correta e se o **driver** do dispositivo foi instalado corretamente:



- e. Na opção **Board** do menu **Tools**, selecione a placa que você está usando clicando sobre a identificação da mesma:



- f. Clique sobre o botão **Upload** (botão redondo com uma seta) para transferir o programa e aguarde o final do processo.



3. Instruções INTER-S para comunicação com Arduino

Na versão experimental INTER-S ARD 2012 foram implementadas instruções que possibilitam a comunicação entre o computador e a placa Arduino. São instruções básicas, mas que permitem explorar de forma simples e rápida os principais recursos do Arduino.

Seguem abaixo as oito instruções básicas:

ARD_COM

Determina o número da porta serial na qual o dispositivo está acoplado.

Sintaxe: **ARD_COM n**

Onde **n** é o número da porta serial.

ARD_Ligapino

Eleva ao nível lógico 1 o pino digital de saída especificado

Sintaxe: **ARD_Ligapino S**

Onde **S** é o número do pino digital de saída. O INTER-S reconhece como saídas os pinos de números 6 a 13.

ARD_Desligapino

Retorna ao nível lógico 0 o pino digital de saída especificado

Sintaxe: **ARD_Desligapino S**

Onde **S** é o número do pino digital de saída. O INTER-S reconhece como saídas os pinos de números 6 a 13.

ARD_Lepino

Lê o estado do pino digital de entrada especificado e coloca o valor de seu nível lógico (0 ou 1) em uma constante interna.

Sintaxe: **ARD_Lepino E**

Onde **E** é o número do pino digital de entrada. O INTER-S reconhece como entradas os pinos de números 2 a 5

As constantes internas, que receberão os valores dos níveis lógicos após a instrução **ARD_Lepino**, são as seguintes:

ard_leitura_2 → armazena o estado lógico do pino digital 2

ard_leitura_3 → armazena o estado lógico do pino digital 3

ard_leitura_4 → armazena o estado lógico do pino digital 4

ard_leitura_5 → armazena o estado lógico do pino digital 5

ARD_Analog

Envia um valor entre 0 e 255 para um pino analógico especificado.

Sintaxe: **ARD_Analog P, V**

Onde **P** é o número do pino analógico de saída e **V** é o valor que será enviado (entre 0 e 255). O INTER-S reconhece como saídas analógicas os pinos de números 9, 10 e 11

ARD_Conectaservo

Configura uma das saídas digitais (pinos de 6 a 13) para controle de um servomotor.

Sintaxe: **ARD_Conectaservo S**

Onde **S** é o número do pino digital no qual o terminal de controle do servomotor está conectado. Pode-se utilizar qualquer um dos pinos de saída reconhecidos pelo INTER-S: pinos de 6 a 13.

ARD_Moveservo

Envia um valor no intervalo de 0 a 180 (ângulo) para o posicionamento de um servomotor.

Sintaxe: **ARD_Moveservo S, A**

Onde **S** é o número do pino digital no qual o terminal de controle do servomotor está conectado e **A** é o ângulo para o posicionamento do servomotor (o ângulo deve estar no intervalo de 0 a 180 graus).

Atrase

Além das instruções relacionadas acima, houve a necessidade de se implementar uma função para provocar um atraso (*delay*) na execução dos programas INTER-S. Esta nova instrução paralisa a execução do programa por um determinado tempo, o qual é especificado em milissegundos.

Sintaxe: **Atrase ms**

Onde **ms** é o tempo em milissegundos.

4. Projetos

Nesta seção apresentaremos alguns projetos básicos com o objetivo de demonstrar o uso das instruções do INTER-S para a comunicação com uma placa Arduino no desenvolvimento de aplicações para automação.

4.1. Projeto 1: Acionamento de um LED

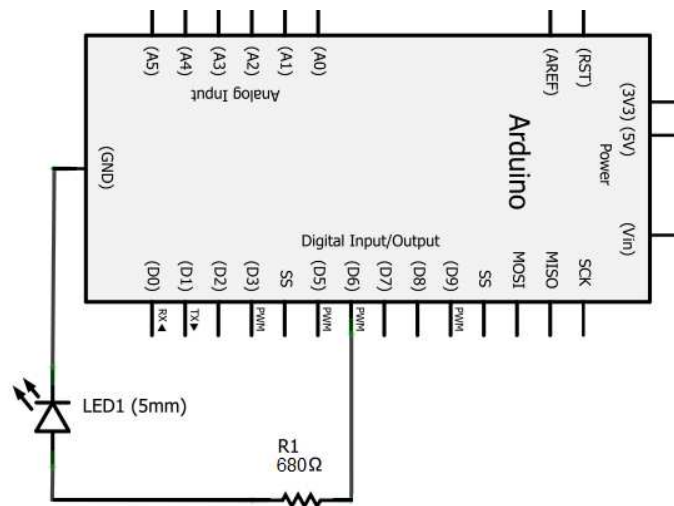
Objetivo:

Construir um programa para acender e apagar um LED através do computador.

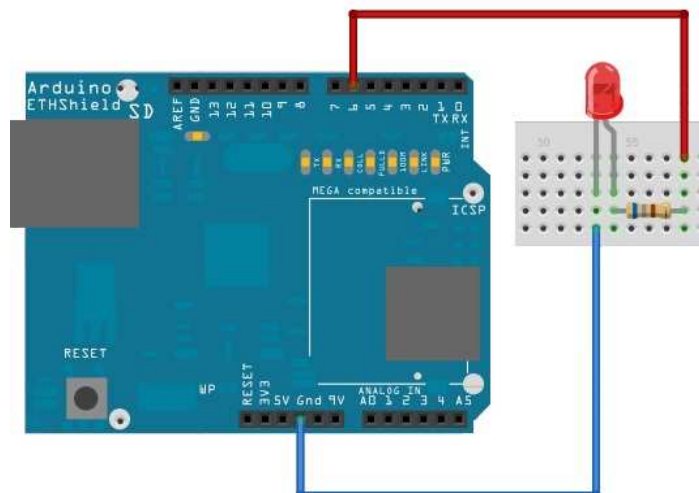
Materiais necessários para a montagem eletrônica:

- 1 LED de 5mm (cor de sua preferência)
- 1 Resistor de 680Ω
- 1 placa de prototipagem Arduino

Diagrama esquemático:



Montagem em matriz de contatos:



Programa INTER-S:

```

Rotina
  Declare x numérico
  ARD_Com 7
  [menu]
  Limpa
  Escreva "PROJETO LED1"
  Escreva "[0] Apagar"
  Escreva "[1] Acender"
  Escreva "[2] Sair"
  Receba "Digite sua opção: ",x
  Se x = 0 Então
    ARD_DesligaPino 6
  FimSe
  Se x = 1 Então
    ARD_LigaPino 6
  FimSe
  Se x # 2 Então
    VaPara menu
  FimSe
FimRotina

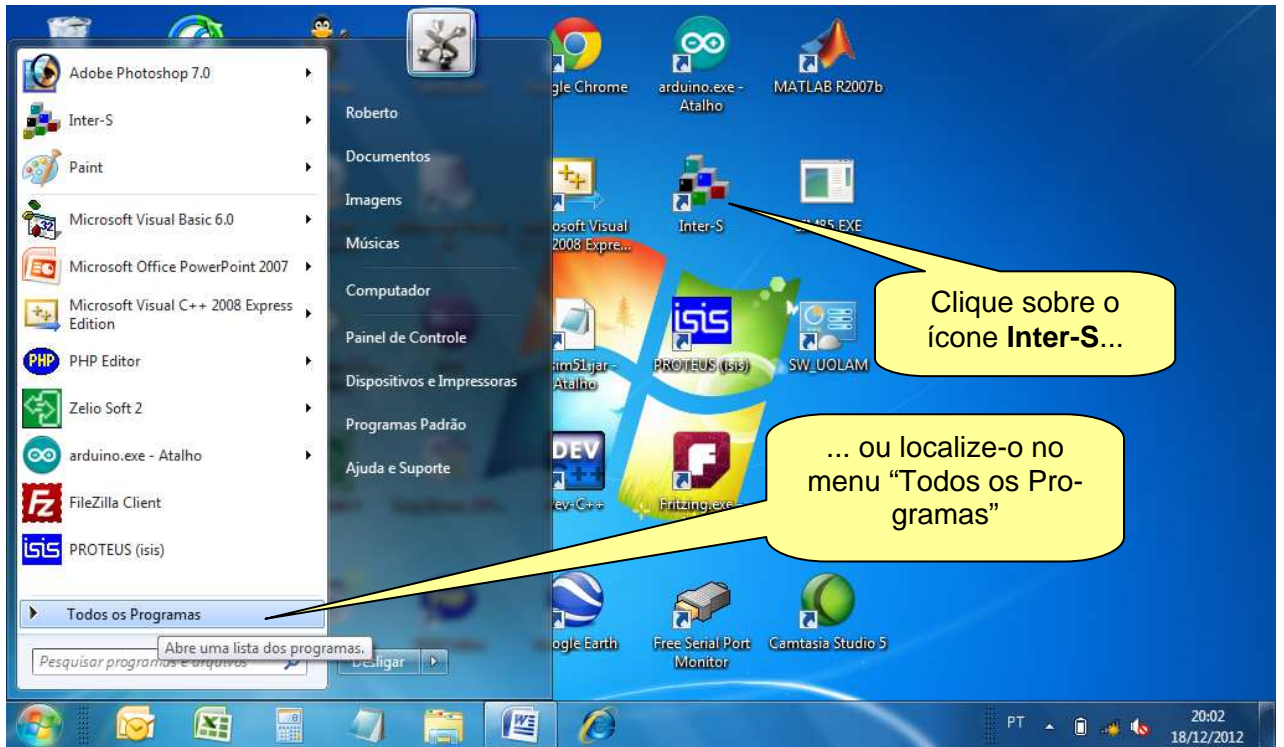
```

Programa INTER-S comentado:

INSTRUÇÕES	COMENTÁRIOS
Rotina	Início do programa
Declare x numérico	Declara a variável x como numérica (esta variável receberá a opção do usuário: 0=apaga o LED; 1=acende o LED; 2=Sai do programa)
ARD_Com 7	Indica o nº da porta serial da placa Arduino (neste caso o nº é 7. Você deve atribuir aqui o nº que o seu sistema gerou para a sua placa Arduino)
[menu]	Marca um ponto de retorno (desvio)
Limpa	Limpa a janela de execução
Escreva "PROJETO LED1"	Escreve na janela a expressão que está entre aspas
Escreva "[0] Apagar"	Escreve na janela a expressão que está entre aspas
Escreva "[1] Acender"	Escreve na janela a expressão que está entre aspas
Escreva "[2] Sair"	Escreve na janela a expressão que está entre aspas
Receba "Digite sua opção: ",x	Coloca na tela a expressão entre aspas e aguarda até que o usuário digite um valor, o qual será atribuído à variável x
Se x = 0 Então	Verifica se x é igual a zero ...
ARD_DesligaPino 6	... caso verdadeiro, desliga o pino 6 do Arduino (apaga o LED)
FimSe	Finaliza a estrutura de decisão
Se x = 1 Então	Verifica se x é igual a um ...
ARD_LigaPino 6	... caso verdadeiro, liga o pino 6 do Arduino (acende o LED)
FimSe	Finaliza a estrutura de decisão
Se x # 2 Então	Verifica se x é diferente de dois...
VaPara menu	... caso verdadeiro, retorna ao ponto [menu]
FimSe	Finaliza a estrutura de decisão
FimRotina	Finaliza o programa

Como construir o programa INTER-S

Com o INTER-S instalado em seu computador, carregue-o clicando sobre o ícone do *desktop* ou use o botão “Iniciar” e localize-o na categoria “Todos os Programas”:

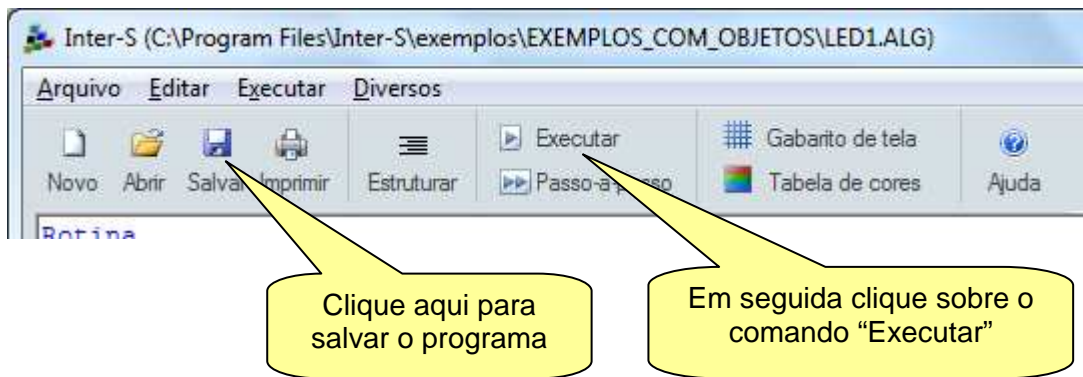


Digite o programa na janela de edição:

```
Inter-S (C:\Program Files\Inter-S\exemplos\EXEMPLOS_COM_OBJETOS\LED1.ALG)
Arquivo  Editar  Executar  Diversos
Novo  Abrir  Salvar  Imprimir  Estruturar  Executar  Gabarito de tela  Tabela de cores  Ajuda
Linha: 3 / Coluna: 13

Rotina
  Declare x numérico
  ARD_Com 7
  [menu]
  Limpa
  Escreva "PROJETO LED1"
  Escreva "[0] Apagar"
  Escreva "[1] Acender"
  Escreva "[2] Sair"
  Receba "Digite sua opção: ",x
  Se x = 0 Então
    ARD_DesligaPino 6
  FimSe
  Se x = 1 Então
    ARD_LigaPino 6
  FimSe
  Se x # 2 Então
    VaPara menu
  FimSe
FimRotina
```

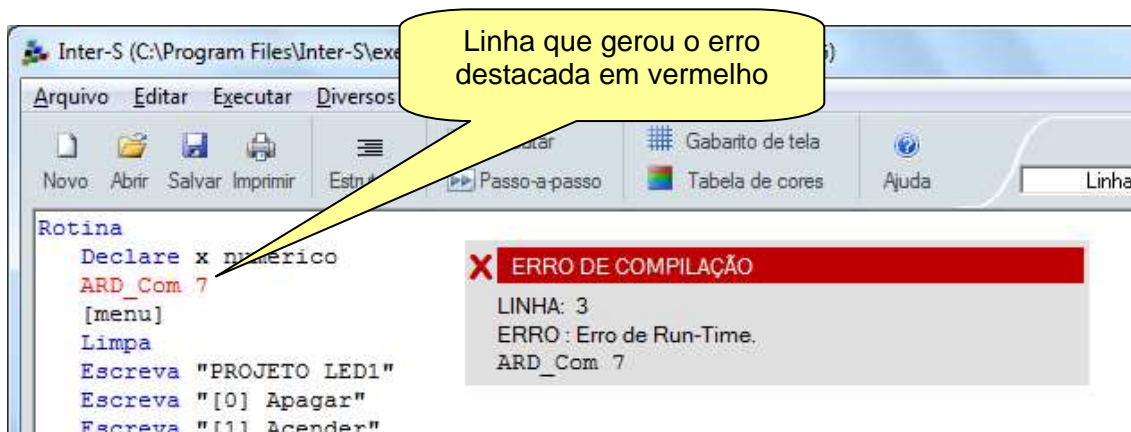
Salve o programa e em seguida execute-o:



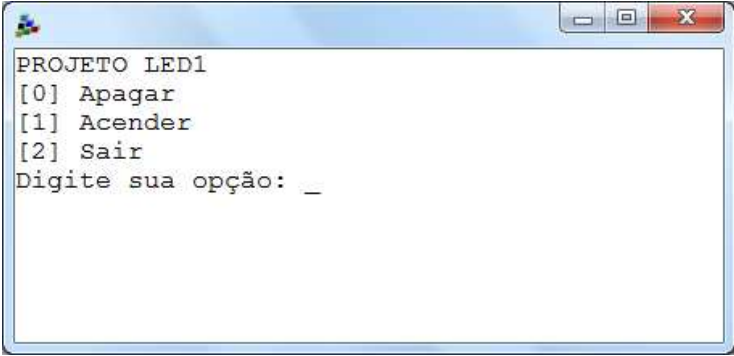
Surgirá a janela de execução onde você deverá clicar sobre a seta para iniciar a aplicação:



No caso de o interpretador INTER-S encontrar algum erro, a execução será abortada e uma mensagem com destaque em vermelho será exibida. Esta mensagem trará também a identificação do erro e a linha que o gerou será destacada em vermelho para que você a corrija. No exemplo abaixo, houve um erro de *Run-Time* na linha 3. Este erro indica que a porta COM7 não existe ou a placa Arduino não está conectada a ela:



Caso o programa não apresente nenhum erro, a tela de execução do nosso primeiro projeto terá o seguinte aspecto:



```
PROJETO LED1
[0] Apagar
[1] Acender
[2] Sair
Digite sua opção: _
```

Observe que será exibido um pequeno menu e o programa ficará aguardando até que o usuário digite uma opção válida.

Se o usuário digitar 0 (zero), será enviado um comando para o Arduino desligar o pino 6, ou seja, levá-lo ao nível zero. Como o anodo do LED está acoplado ao pino 6, esta operação fará com que ele apague.

Se o usuário digitar 1 (um), será enviado o comando para levar o pino 6 ao nível 1, o que fará o LED acender.

Caso o usuário digite 2 (dois), o programa será encerrado.

Qualquer outro valor diferente de 2 fará com que o menu seja exibido novamente e o programa continuará aguardando a opção do usuário.

Sugestões:

Analise com mais profundidade o programa INTER-S deste projeto e promova algumas alterações, como por exemplo: incluir uma nova opção para que o LED pisque 10 vezes.

Você também pode acrescentar outro LED utilizando a saída digital 7 e modificar o programa para controlar os dois.

Para ilustrar as possibilidades, o programa abaixo faz com que o LED conectado ao pino 6 pisque 10 vezes, sendo que o intervalo entre cada piscada é de 1 segundo, ou seja, o LED permanece aceso durante 500 milissegundos (meio segundo) e apagado durante o mesmo tempo. Veja que as instruções para ligar e desligar o pino 6 estão dentro de uma estrutura de repetição cujo número de ciclos é 10 (Para $i = 1$ até 10...). Para determinar o tempo de duração de cada ação, empregamos a instrução **Atrase 500**, a qual provoca uma pausa de 500 milissegundos:

```
Rotina
  Declare i numérico
  ARD_Com 7
  Para i = 1 Até 10 Faça
    ARD_LigaPino 6; Atrase 500
    ARD_DesligaPino 6; Atrase 500
  FimPara
FimRotina
```


4.2. Projeto 2: Sequência de 8 LEDs

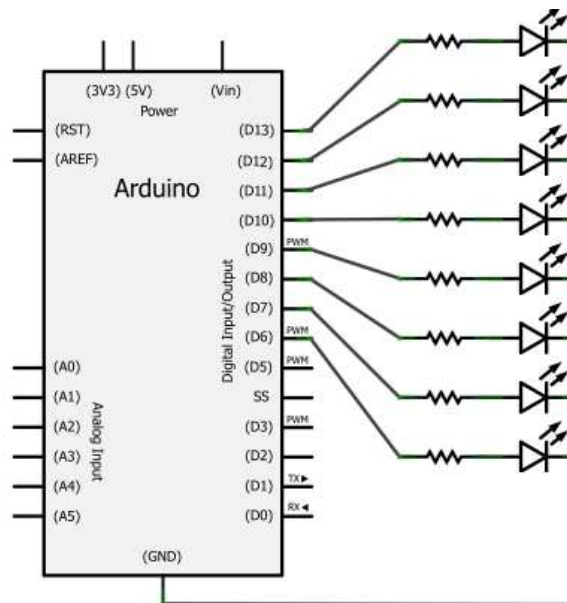
Objetivo:

Construir um programa para acender e apagar sequencialmente um conjunto de 8 LEDs.

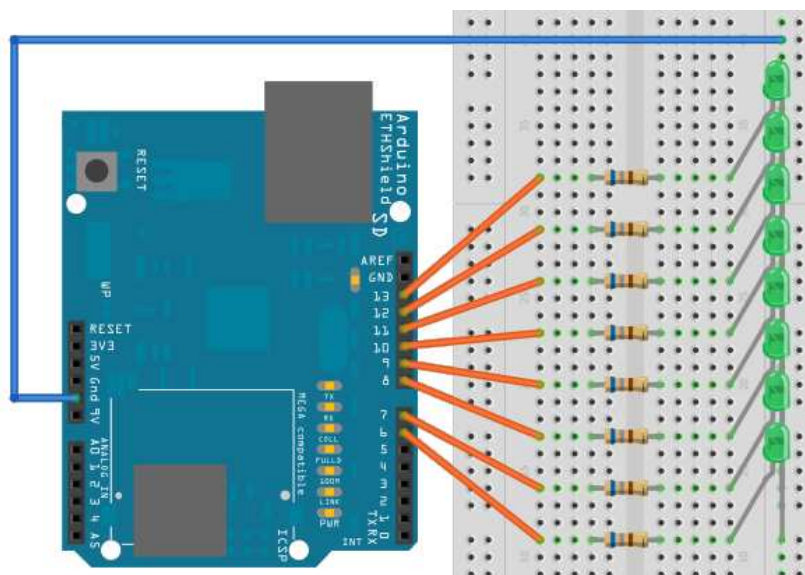
Materiais necessários para a montagem eletrônica:

- 8 LEDs de 3mm
- 8 Resistores de 680Ω
- 1 matriz de contatos
- 1 placa de prototipagem Arduino

Diagrama esquemático:



Montagem em matriz de contatos:



Programa INTER-S:

```

Rotina
  Declare i,j numéricos
  ARD_Com 7
  Para i = 1 Até 10 Faça
    Para j = 6 Até 13 Faça
      ARD_LigaPino j; Atrase 50
      ARD_DesligaPino j; Atrase 50
    FimPara
  FimPara
FimRotina

```

Programa INTER-S comentado:

INSTRUÇÕES	COMENTÁRIOS
Rotina	Início do programa
Declare i,j numéricos	Declara as variáveis i e j como numéricas (estas variáveis serão utilizadas como contadoras de ciclos nas estruturas de repetição "PARA")
ARD_Com 7	Indica o nº da porta serial da placa Arduino (neste caso o nº é 7. Você deve atribuir aqui o nº que o seu sistema gerou para a sua placa Arduino)
Para i = 1 Até 10 Faça	Inicia uma estrutura de repetição onde a variável i assumirá o papel de contadora de ciclos, iniciando em 1 e terminando em 10 (serão realizados 10 ciclos).
Para j = 6 Até 13 Faça	Inicia outra estrutura de repetição onde a variável j assumirá o papel de contadora de ciclos, iniciando em 6 e terminando em 13 (serão realizados 8 ciclos). Observe que esta estrutura encontra-se dentro da estrutura anterior, portanto, ela será executada 10 vezes.
ARD_LigaPino j; Atrase 50	Liga o pino j do Arduino (ascende o LED conectado ao pino representado pela variável j). A variável j realiza a contagem dos ciclos da estrutura de repetição e assume os valores de 6 a 13 (que são os pinos onde os 8 LEDs estão conectados). Após a instrução ARD_Ligapino, é realizada uma pausa de 50 milissegundos (tempo em que o LED permanecerá aceso).
ARD_DesligaPino j; Atrase 50	Desliga o pino j (apaga o LED correspondente) e realiza uma pausa de 50 milissegundos.
FimPara	Instrução que indica o final da última estrutura de repetição aberta. Esta instrução verifica o valor da variável j: se j ainda não atingiu o limite (que neste caso é 13), o valor de j será incrementado em 1 e um novo ciclo será realizado.
FimPara	Instrução que indica o final da primeira estrutura de repetição aberta. Esta instrução verifica o valor da variável i: se i ainda não atingiu o limite (que neste caso é 10), o valor de i será incrementado em 1 e um novo ciclo será realizado.
FimRotina	Finaliza o programa

Sugestões:

O programa deste projeto faz com que os LEDs acoplados nos pinos de 6 a 13 acendam e apaguem sequencialmente. A estrutura de repetição iniciada pela instrução “Para i=1 Até 10 Faça” determina o número de vezes que a sequência será realizada (1 a 10) e a estrutura iniciada pela instrução “Para j = 6 Até 13 Faça” determina a sequência de acionamento dos LEDs.

Tente alterar os parâmetros destas estruturas e veja o resultado.

O programa abaixo é outro exemplo de aplicação para esta montagem. Ele permite que o usuário escolha o LED que será acionado:

Rotina

```
Declare x numérico
```

```
ARD_Com 7
```

```
[selecao]
```

```
Limpa
```

```
Escreva "SELEÇÃO DE PINO"
```

```
Receba "Informe o pino a ser ligado (6 a 13)...: ",x
```

```
ARD_LigaPino x
```

```
Receba "Informe o pino a ser desligado (6 a 13): ",x
```

```
ARD_DesligaPino x
```

```
VaPara selecao
```

FimRotina

4.3. Projeto 3: Display de 7 segmentos

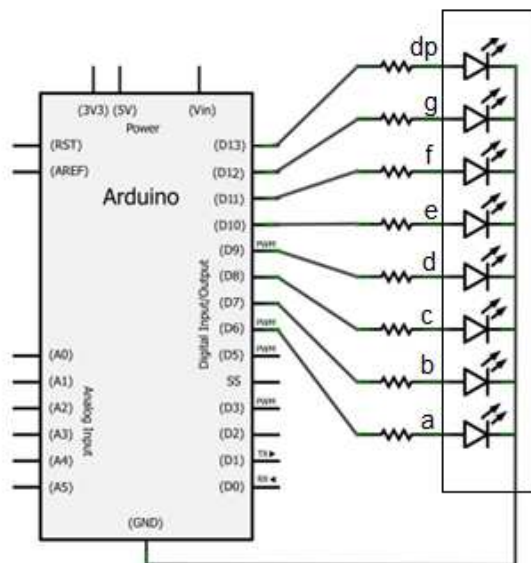
Objetivo:

Construir um programa para acender e apagar os segmentos de um display. Este projeto é uma adaptação do projeto 2, onde os 8 LEDs serão substituídos por um display de 7 segmentos de catodo comum.

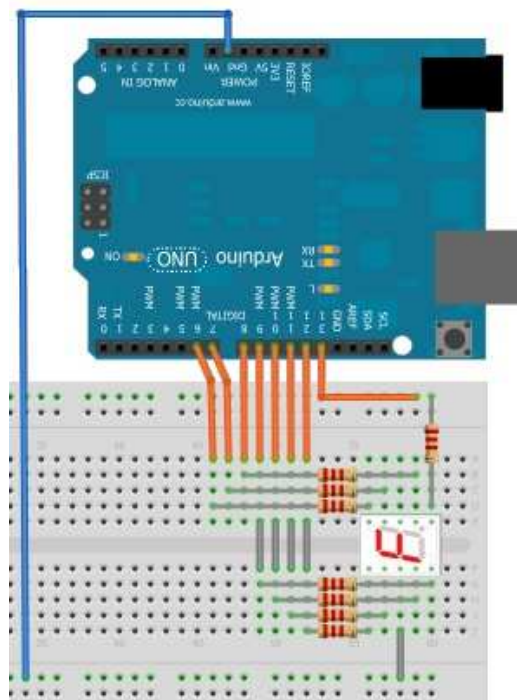
Materiais necessários para a montagem eletrônica:

- 1 display de 7 segmentos – catodo comum
- 8 Resistores de $2K2\Omega$
- 1 matriz de contatos
- 1 placa de prototipagem Arduino

Diagrama esquemático:

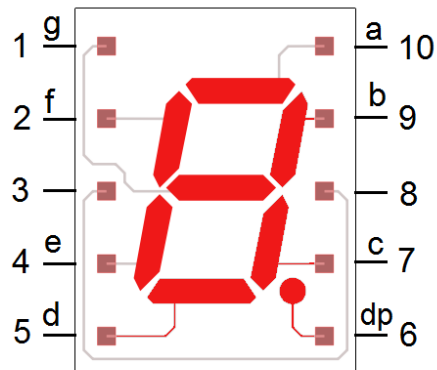


Montagem em matriz de contatos:



Pinagem do display:

Para este projeto foi utilizado um display da Kingbright modelo SC39-11 cuja pinagem é mostrada na figura abaixo:



Observe que cada segmento é identificado por uma letra de **a** à **g** e associado a um dos pinos (1 a 10). Os pinos 3 e 8 correspondem ao terra (GND) e o pino 6 corresponde ao ponto (dp).

Caso você utilize um modelo diferente, consulte a folha de dados (*datasheet*) correspondente ao modelo, pois a pinagem pode apresentar outra configuração.

Conexão dos pinos:

A tabela abaixo mostra a correspondência entre os pinos do display e os pinos do Arduino (considerando o modelo SC39-11 da Kingbright). Faça os ajustes necessários de acordo com o modelo que você usará:

Pino do display SC39-11	Pino do Arduino
1 (segmento g)	Pino digital 12
2 (segmento f)	Pino digital 11
3 ou 8 (GND – terra)	Pino GND – terra
4 (segmento e)	Pino digital 10
5 (segmento d)	Pino digital 9
6 (dp – ponto)	Pino digital 13
7 (segmento c)	Pino digital 8
9 (segmento b)	Pino digital 7
10 (segmento a)	Pino digital 6

Programa INTER-S:

Para testar sua montagem, utilize os mesmos programas propostos no projeto 2 (sequência de 8 de LEDs).

Exemplo de um programa mais sofisticado para testar a montagem

O programa abaixo utiliza recursos avançados do INTER-S para montar uma interface que simula um display. Esta interface permite que o usuário clique sobre um dos seguimentos para acendê-lo ou apagá-lo:

Rotina

```
Declare bt1 BotãoDeComando(2,5,2,13,"a")
Declare bt2 BotãoDeComando(3,14,6,16,"b")
Declare bt3 BotãoDeComando(9,14,12,16,"c")
Declare bt4 BotãoDeComando(14,5,14,13,"d")
Declare bt5 BotãoDeComando(9,2,12,4,"e")
Declare bt6 BotãoDeComando(3,2,6,4,"f")
Declare bt7 BotãoDeComando(8,5,8,13,"g")
Declare flag1,flag2,flag3,flag4,flag5,flag6,flag7 numéricos
ARD_Com 7
[leitura]
AtivaBotão
CorTexto branco
Se bt1 = 1 Então
  Se flag1 = 0 Então
    ARD_LigaPino 6;flag1=1; CorBotão bt1,vermelho+
    Senão
      ARD_DesligaPino 6;flag1=0; CorBotão bt1,branco
  FimSe
FimSe
Se bt2 = 1 Então
  Se flag2 = 0 Então
    ARD_LigaPino 7;flag2=1; CorBotão bt2,vermelho+
    Senão
      ARD_DesligaPino 7;flag2=0; CorBotão bt2,branco
  FimSe
FimSe
Se bt3 = 1 Então
  Se flag3 = 0 Então
    ARD_LigaPino 8;flag3=1; CorBotão bt3,vermelho+
    Senão
      ARD_DesligaPino 8;flag3=0; CorBotão bt3,branco
  FimSe
FimSe
Se bt4 = 1 Então
  Se flag4 = 0 Então
    ARD_LigaPino 9;flag4=1; CorBotão bt4,vermelho+
    Senão
      ARD_DesligaPino 9;flag4=0; CorBotão bt4,branco
  FimSe
FimSe
Se bt5 = 1 Então
  Se flag5 = 0 Então
    ARD_LigaPino 10;flag5=1; CorBotão bt5,vermelho+
    Senão
      ARD_DesligaPino 10;flag5=0; CorBotão bt5,branco
  FimSe
FimSe
```

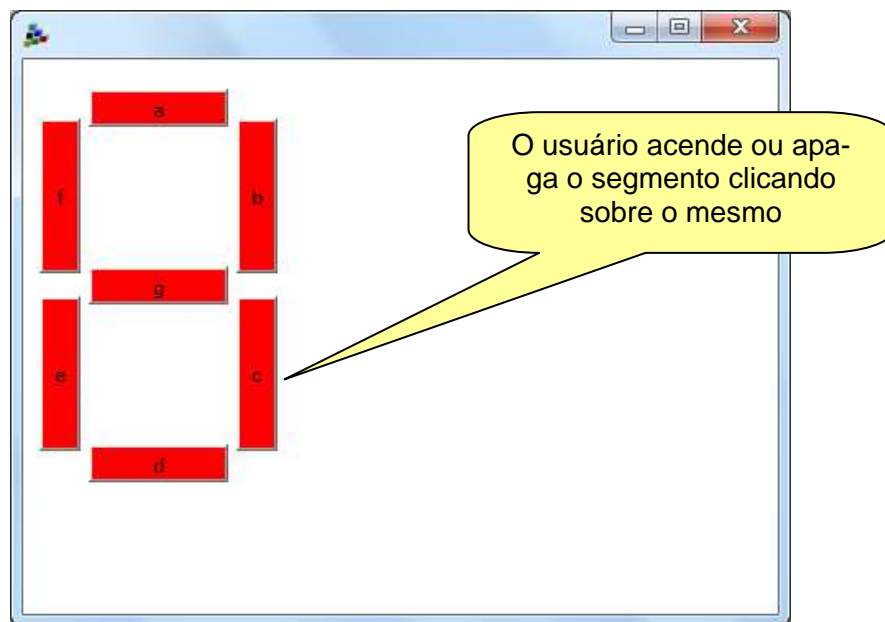


```

Se bt6 = 1 Então
  Se flag6 = 0 Então
    ARD_LigaPino 11;flag6=1; CorBotão bt6,vermelho+
  Senão
    ARD_DesligaPino 11;flag6=0; CorBotão bt6,branco
  FimSe
FimSe
Se bt7 = 1 Então
  Se flag7 = 0 Então
    ARD_LigaPino 12;flag7=1; CorBotão bt7,vermelho+
  Senão
    ARD_DesligaPino 12;flag7=0; CorBotão bt7,branco
  FimSe
FimSe
VaPara leitura
FimRotina

```

A tela de execução deste programa terá o seguinte aspecto:



4.4. Projeto 4: Controle de um servomotor

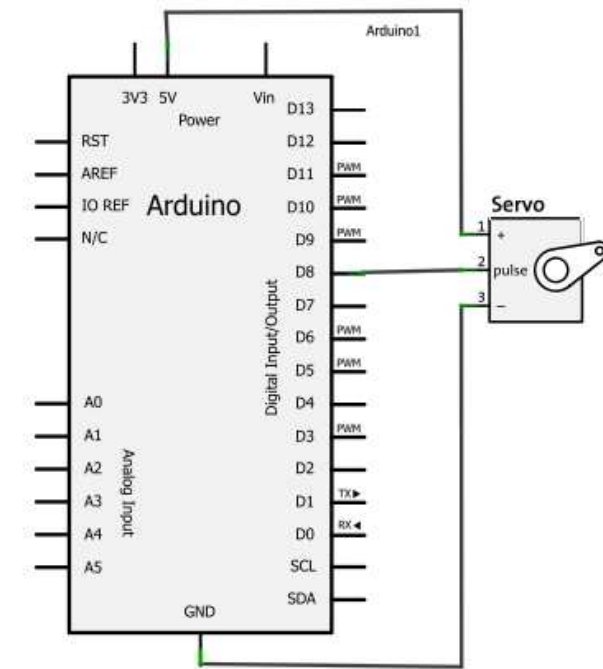
Objetivo:

Construir um programa para controlar um servomotor através do computador.

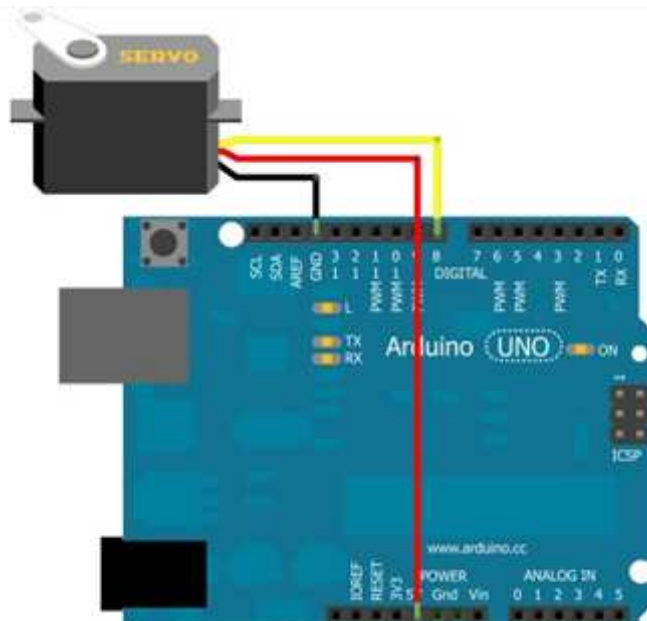
Materiais necessários para a montagem eletrônica:

- 1 servomotor pequeno com rotação de 180° / voltagem de operação de aproximadamente 5V / padrão: 3 terminais.
- 1 placa de prototipagem Arduino

Diagrama esquemático:



Montagem:



Os terminais de um servo motor normalmente têm a seguinte configuração:

- Vermelho: alimentação (+Vcc)
- Preto: Terra (GND)
- Amarelo ou laranja: controle

O terminal vermelho deve ser conectado ao pino 5V do Arduino. O terminal preto deve ser conectado ao pino GND e o terminal amarelo (ou laranja) deve ser conectado ao pino digital 6 (o qual será controlado pelo programa INTER-S)

Programa INTER-S:

```

Rotina
  Declare a numérico
  ARD_Com 7
  ARD_ConectaServo 6
  [menu]
  Limpa
  Escreva "PROJETO SERV01"
  Receba "Digite o ângulo: ",a
  ARD_MoveServo 6,a
  Vapara menu
FimRotina
  
```

Programa INTER-S comentado:

INSTRUÇÕES	COMENTÁRIOS
Rotina	Início do programa
Declare a numérico	Declara a variável a como numérica (esta variável receberá o ângulo digitado pelo usuário)
ARD_Com 7	Indica o nº da porta serial da placa Arduino (neste caso o nº é 7. Você deve atribuir aqui o nº que o seu sistema gerou para a sua placa Arduino)
ARD_ConectaServo 6	Indica o nº do pino digital do Arduino onde o servo está conectado
[menu]	Indica uma marca (ponto de retorno)
Limpa	Limpa a tela
Escreva "PROJETO SERV01"	Escreve na janela a expressão que está entre aspas
Receba "Digite o ângulo: ",a	Coloca na tela a expressão que está entre aspas e aguarda até que o usuário digite um valor, o qual será atribuído à variável a
ARD_MoveServo 6,a	Instrução que envia o valor da variável a (ângulo) para o posicionamento do servomotor conectado ao pino 6.
Vapara menu	Desvia o processamento para o ponto de retorno [menu]
FimRotina	Finaliza o programa

Sugestão:

O programa INTER-S apresentado na página anterior, quando em execução, exibirá na tela a mensagem “Digite o ângulo” e aguardará até que o usuário digite um valor.

O valor digitado pelo usuário será armazenado na variável **a**, a qual, na instrução subsequente, será utilizada como parâmetro para o posicionamento do servomotor.

Observe que o programa não faz nenhum tratamento em relação ao intervalo do valor digitado, ou seja, mesmo que o usuário informe um valor fora da faixa de 0 a 180 o programa enviará este valor para o Arduino (lembrando que o servomotor usado neste projeto opera apenas na rotação de 180°). Portanto, como sugestão, inclua uma estrutura de decisão em seu programa (ou duas) para verificar se o valor informado pelo usuário está dentro da faixa de 0 a 180.

Exemplo de um programa mais sofisticado para testar a montagem

O programa abaixo utiliza os recursos gráficos do INTER-S para gerar uma interface mais sofisticada que mostra em uma janela gráfica a posição do rotor do servo simultaneamente ao posicionamento real:

Rotina

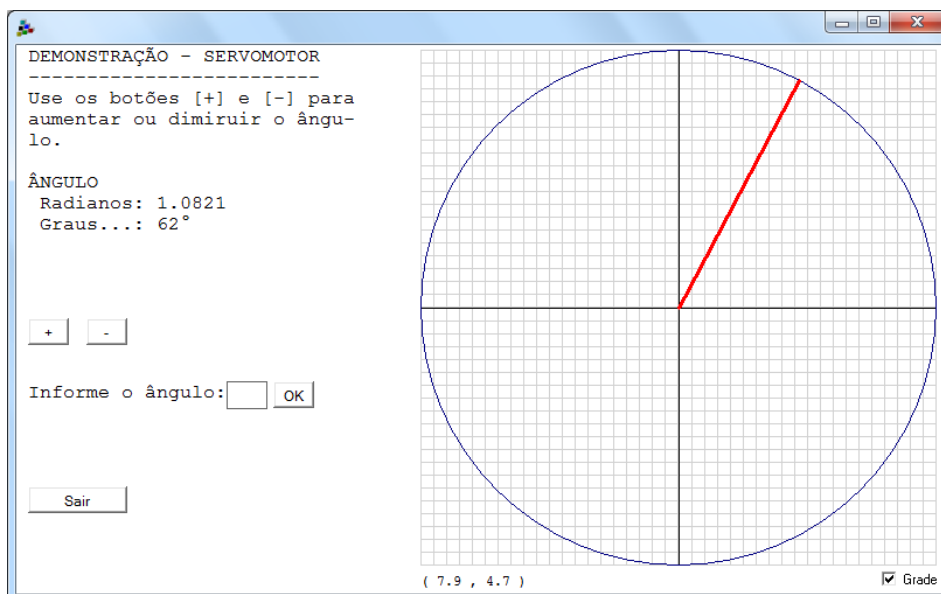
```
Declare pi, px, py, angulo, raio, gard, incremento numéricos
Declare botaomais BotãoDeComando(14,2,14,5,"+")
Declare botaomenos BotaoDeComando(14,7,14,10,"-")
Declare botaosair BotaoDeComando(22,2,22,10,"Sair")
Declare caixaangulo CaixaDeTexto(17,19,17,22) numérica
Declare botaoangulo BotãoDeComando(17,23,17,26,"OK")
pi=3.14159; incremento=(2*pi/360)*2; raio=10; angulo=0
Escala 10; Gráfico; HabilitaGrade; TamanhoPonto 1
ARD_Com 7
ARD_ConectaServo 6
ARD_MoveServo 6, 0
Processe tela
[volta]
AtivaBotao
Se botaosair = 1 Então
    Interrompa
FimSe
Se botaomais = 1 Então
    angulo = angulo+incremento; Processe tela
FimSe
Se botaomenos = 1 Então
    angulo = angulo-incremento; Processe tela
FimSe
Se botaoangulo = 1 Então
    Se caixaangulo > = 0 e caixaangulo < = 180 Então
        angulo=caixaangulo*pi/180; Processe tela
    FimSe
FimSe
VaPara volta
FimRotina
```

```

SubRotina tela
  LimpaGráfico
  CorPonto azul
  Círculo 0,0,raio
  Se angulo < 0 Então
    angulo=0
  FimSe
  Se angulo > pi Então
    angulo=pi
  FimSe
  px=cos(angulo)*raio
  py=seno(angulo)*raio
  CorPonto vermelho+
  TamanhoPonto 3
  Linha 0,0/px,py
  TamanhoPonto 1
  Limpa
  Escreva " DEMONSTRAÇÃO - SERVOMOTOR"
  Escreva " -----"
  Escreva " Use os botões [+] e [-] para"
  Escreva " aumentar ou diminuir o ângu-"
  Escreva " lo."
  Escreva " "
  Escreva " ÂNGULO"
  Escreva " Radianos: ", arred4(angulo)
  Escreva " Graus...: ", trunca(angulo*180/pi),"°"
  Escreva " "
  Posicione 17,2; Escreva "Informe o ângulo:"
  gard = trunca(angulo*180/pi)
  Se gard > 180 Então
    ARD_MoveServo 6, 180
  Senão
    ARD_MoveServo 6, gard
  FimSe
FimSubRotina

```

A tela de execução deste programa terá o seguinte aspecto:



4.5. Projeto 5: Braço robótico (controle de 4 servos)

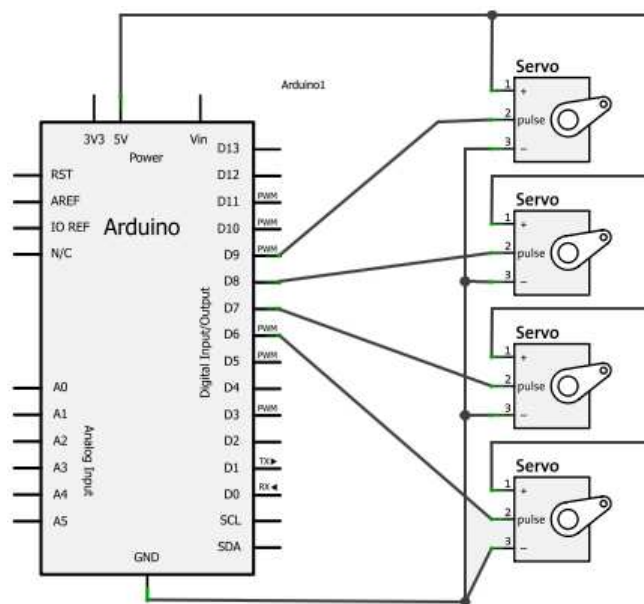
Objetivo:

Construir um programa para controlar quatro servomotores através do computador. Este projeto é uma extensão do projeto 4 e tem como propósito oferecer uma alternativa para o controle de mecanismos microcontrolados que utilizam mais de um atuador.

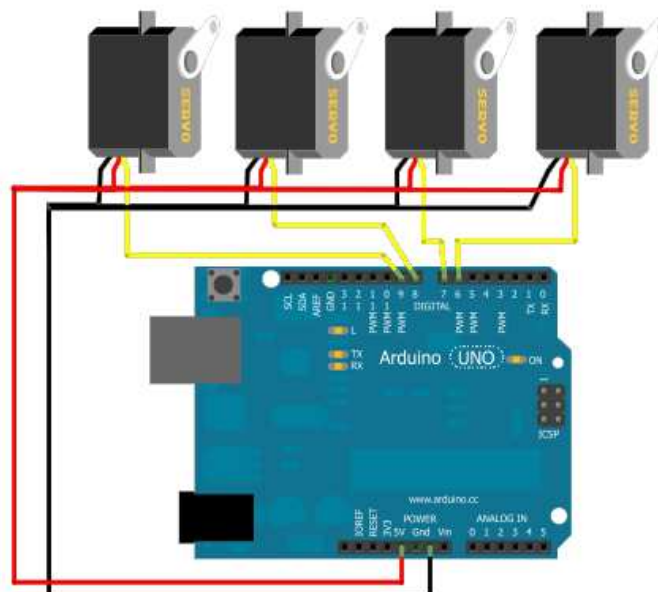
Materiais necessários para a montagem eletrônica:

- 4 servomotores pequenos com rotação de 180° / voltagem de operação de aproximadamente 5V / padrão: 3 terminais.
- 1 placa de prototipagem Arduino

Diagrama esquemático:



Montagem:



Programa INTER-S:

Rotina

```

Declare a,s numéricos
ARD_Com 7
ARD_ConectaServo 6
ARD_ConectaServo 7
ARD_ConectaServo 8
ARD_ConectaServo 9
[menu]
Limpa
Escreva "PROJETO 4 SERVOS"
Receba "Número do servo (6 a 9): ",s
Receba "Ângulo.....: ",a
Se s < 6 ou s > 9 Então
    Escreva "Servo inválido!"
    Pausa
    VaPara menu
FimSe
Se a < 0 ou a > 180 Então
    Escreva "Ângulo inválido!"
    Pausa
    VaPara menu
FimSe
ARD_MoveServo s,a
VaPara menu
FimRotina

```

Programa INTER-S comentado:

INSTRUÇÕES	COMENTÁRIOS
Rotina	Início do programa
Declare a,s numéricos	Declara as variáveis a e s como numéricas (a variável a receberá o ângulo digitado pelo usuário e a variável s receberá o número do servo)
ARD_Com 7	Indica o nº da porta serial da placa Arduino (neste caso o nº é 7. Você deve atribuir aqui o nº que o seu sistema gerou para a sua placa Arduino)
ARD_ConectaServo 6	Atribui o pino 6 para o 1º servo
ARD_ConectaServo 7	Atribui o pino 7 para o 2º servo
ARD_ConectaServo 8	Atribui o pino 8 para o 3º servo
ARD_ConectaServo 9	Atribui o pino 9 para o 4º servo
[menu]	Indica uma marca (ponto de retorno)
Limpa	Limpa a tela
Escreva "PROJETO SERVOS"	Escreve na janela a expressão que está entre aspas
Receba "Número do servo (6 a 9):",s	Coloca na tela a expressão que está entre aspas e aguarda até que o usuário digite um valor, o qual será atribuído à variável s
Receba "Ângulo.....: ",a	Coloca na tela a expressão que está entre aspas e aguarda até que o usuário digite um valor, o qual será atribuído à variável a

<code>Se s < 6 ou s > 9 Então</code>	Verifica se o valor da variável s está fora do intervalo de 6 a 9 (a variável s deve representar o nº do pino digital onde está conectado um dos servos)
<code>Escreva "Servo inválido!"</code>	... caso positivo, ou seja, se a variável s estiver fora do intervalo, o programa escreverá na tela a mensagem "Servo inválido!"
<code>Pausa</code>	... dará uma pausa ...
<code>VaPara menu</code>	... e retornará ao ponto marcado como [menu], ou seja, voltará ao ponto onde o programa solicita os valores das variáveis.
<code>FimSe</code>	Finaliza a estrutura de decisão
<code>Se a < 0 ou a > 180 Então</code>	Verifica se o valor da variável a está fora do intervalo de 0 a 180 (a variável a representa o ângulo para o posicionamento do servo)
<code>Escreva "Ângulo inválido!"</code>	... caso positivo, ou seja, se a variável a estiver fora do intervalo, o programa escreverá na tela a mensagem "Ângulo inválido!"
<code>Pausa</code>	... dará uma pausa ...
<code>VaPara menu</code>	... e retornará ao ponto marcado como [menu], ou seja, voltará ao ponto onde o programa solicita os valores das variáveis.
<code>FimSe</code>	Finaliza a estrutura de decisão
<code>ARD_MoveServo s,a</code>	Envia para a placa Arduino a instrução para que o servo conectado ao pino s seja posicionado no ângulo a (lembrando que s representa o nº do servo informado pelo usuário e a representa o ângulo também informado pelo usuário)
<code>Vapara menu</code>	Desvia o processamento para o ponto de retorno [menu]
<code>FimRotina</code>	Finaliza o programa

Exemplo de um programa mais sofisticado para testar a montagem

O programa abaixo é uma proposta para a construção de uma interface gráfica amigável onde o usuário poderá controlar os quatro servos utilizando o mouse. A interface também oferece recursos como a visualização simultânea em tela das posições dos motores e um botão de comando que reproduz automaticamente todos os movimentos anteriores realizados pelo usuário:

Rotina

```

Declare s1mais BotãoDeComando(5,10,5,16,"Servo1 +")
Declare s1menos BotãoDeComando(5,18,5,24,"Servo1 -")
Declare s2mais BotãoDeComando(7,10,7,16,"Servo2 +")
Declare s2menos BotãoDeComando(7,18,7,24,"Servo2 -")
Declare s3mais BotãoDeComando(9,10,9,16,"Servo3 +")
Declare s3menos BotãoDeComando(9,18,9,24,"Servo3 -")
Declare s4mais BotãoDeComando(11,10,11,16,"Servo4 +")
Declare s4menos BotãoDeComando(11,18,11,24,"Servo4 -")
Declare executa BotãoDeComando(14,10,14,16,"REPETE")
Declare limpeza BotãoDeComando(14,18,14,24,"LIMPA")
Declare decremento BotãoDeComando(16,10,16,16,"Passo -")
Declare incremento BotãoDeComando(16,18,16,24,"Passo +")

Declare pi, rad, raio, inc numéricos
Declare px, py numéricos
Declare a1, a2, a3, a4, i numéricos

```

```

Declare servoanterior, servoatual numéricos
Declare anguloanterior, anguloatual, ind numéricos
Declare Arranjo servo[100] numérico
Declare Arranjo angulo[100] numérico

CorBotão s1mais,prata ; CorBotão s1menos,prata
CorBotão s2mais,prata ; CorBotão s2menos,prata
CorBotão s3mais,prata ; CorBotão s3menos,prata
CorBotão s4mais,prata ; CorBotão s4menos,prata
CorBotão executa,prata
CorBotão limpeza,prata
CorBotão incremento, prata
CorBotão decremento, prata

pi=3.14159; raio=4; inc=4
a1=90; a2=90; a3=90; a4=90
servoanterior=0; servoatual=0
anguloanterior=0; anguloatual=0
ind=0

ARD_Com 7
ARD_ConectaServo 6
ARD_ConectaServo 7
ARD_ConectaServo 8
ARD_ConectaServo 9
ARD_MoveServo 6,90
ARD_MoveServo 7,90
ARD_MoveServo 8,90
ARD_MoveServo 9,90

Gráfico
Processe tela
Processe plota

[inicio]

AtivaBotão

Se s1mais = 1 Então
    a1=a1+inc
    Se a1 > 180 Então
        a1=180
    FimSe
    ARD_MoveServo 6,a1 ; servoatual=6; anguloatual=a1
    Processe tela
    Processe plota
FimSe
Se s1menos = 1 Então
    a1=a1-inc
    Se a1 < 0 Então
        a1=0
    FimSe
    ARD_MoveServo 6,a1 ; servoatual=6; anguloatual=a1
    Processe tela
    Processe plota
FimSe

```

```

Se s2mais = 1 Então
  a2=a2+inc
  Se a2 > 180 Então
    a2=180
  FimSe
  ARD_MoveServo 7,a2 ; servoatual=7; anguloatual=a2
  Processe tela
  Processe plota
FimSe
Se s2menos = 1 Então
  a2=a2-inc
  Se a2 < 0 Então
    a2=0
  FimSe
  ARD_MoveServo 7,a2 ; servoatual=7; anguloatual=a2
  Processe tela
  Processe plota
FimSe
Se s3mais = 1 Então
  a3=a3+inc
  Se a3 > 180 Então
    a3=180
  FimSe
  ARD_MoveServo 8,a3 ; servoatual=8; anguloatual=a3
  Processe tela
  Processe plota
FimSe
Se s3menos = 1 Então
  a3=a3-inc
  Se a3 < 0 Então
    a3=0
  FimSe
  ARD_MoveServo 8,a3 ; servoatual=8; anguloatual=a3
  Processe tela
  Processe plota
FimSe
Se s4mais = 1 Então
  a4=a4+inc
  Se a4 > 180 Então
    a4=180
  FimSe
  ARD_MoveServo 9,a4 ; servoatual=9; anguloatual=a4
  Processe tela
  Processe plota
FimSe
Se s4menos = 1 Então
  a4=a4-inc
  Se a4 < 0 Então
    a4=0
  FimSe
  ARD_MoveServo 9,a4 ; servoatual=9; anguloatual=a4
  Processe tela
  Processe plota
FimSe

```

```

Se executa = 1 Então
  a1=90; a2=90; a3=90; a4=90
  ARD_MoveServo 6,90
  ARD_MoveServo 7,90
  ARD_MoveServo 8,90
  ARD_MoveServo 9,90
  Processe tela
  Para i = 1 Até ind Faça
    ARD_MoveServo servo[i],angulo[i]
    Caso servo[i]
      Seja = 6 Faça a1 = angulo[i]
      Seja = 7 Faça a2 = angulo[i]
      Seja = 8 Faça a3 = angulo[i]
      Seja = 9 Faça a4 = angulo[i]
    FimCaso
    Processe tela
    Processe plota
    Atrase 500
  FimPara
FimSe
Se limpeza = 1 Então
  a1=90; a2=90; a3=90; a4=90
  servoanterior=0; servoatual=0
  anguloanterior=0; anguloatual=0
  ind=0
  ARD_MoveServo 6,90
  ARD_MoveServo 7,90
  ARD_MoveServo 8,90
  ARD_MoveServo 9,90
  Processe tela
  Processe plota
FimSe

Se incremento = 1 Então
  inc=inc+1
  Se inc > 30 Então
    inc=30
  FimSe
  Processe TELA
FimSe

Se decremento = 1 Então
  inc=inc-1
  Se inc < 2 Então
    inc=1
  FimSe
  Processe tela
FimSe

VaPara inicio

Pausa
FimRotina

```

```

SubRotina tela
  CorTexto oliva
  Moldura 3,6,12,30, amarelo
  CorTexto preto
  Posicione 3,14; Escreva "CONTROLE"
  Posicione 5,26; Escreva a1,"°"
  Posicione 7,26; Escreva a2,"°"
  Posicione 9,26; Escreva a3,"°"
  Posicione 11,26; Escreva a4,"°"
  CorTexto branco
  Moldura 16,26,16,34,branco
  CorTexto preto
  Posicione 16,26; Escreva inc,"°"
  Se servoatual # servoanterior e ind < 100 Então
    ind=ind+1
    servo[ind]=servoanterior
    angulo[ind]=anguloanterior
  FimSe
  servoanterior=servoatual
  anguloanterior=anguloatual
FimSubRotina

```

```

SubRotina plota
  LimpaGráfico
  TamanhoPonto 1
  CorPonto ciano
  Círculo -5,5,3
  Círculo -5,-5,3
  Círculo 5,5,3
  Círculo 5,-5,3

  TamanhoPonto 4
  CorPonto vermelho+; Círculo -5,5,1
  CorPonto azul+; Círculo 5,5,1
  CorPonto verde; Círculo -5,-5,1
  CorPonto preto; Círculo 5,-5,1

  rad=a1*3.14159/180
  px=cos(rad)*raio; py=seno(rad)*raio
  CorPonto vermelho+
  px=px-5;py=py+5
  Linha -5,5/px,py

  rad=a2*pi/180
  px=cos(rad)*raio; py=seno(rad)*raio
  CorPonto azul+
  px=px+5; py=py+5
  Linha 5,5/px,py

  rad=a3*pi/180
  px=cos(rad)*raio; py=seno(rad)*raio
  CorPonto verde
  px=px-5;py=py-5
  Linha -5,-5/px,py

```



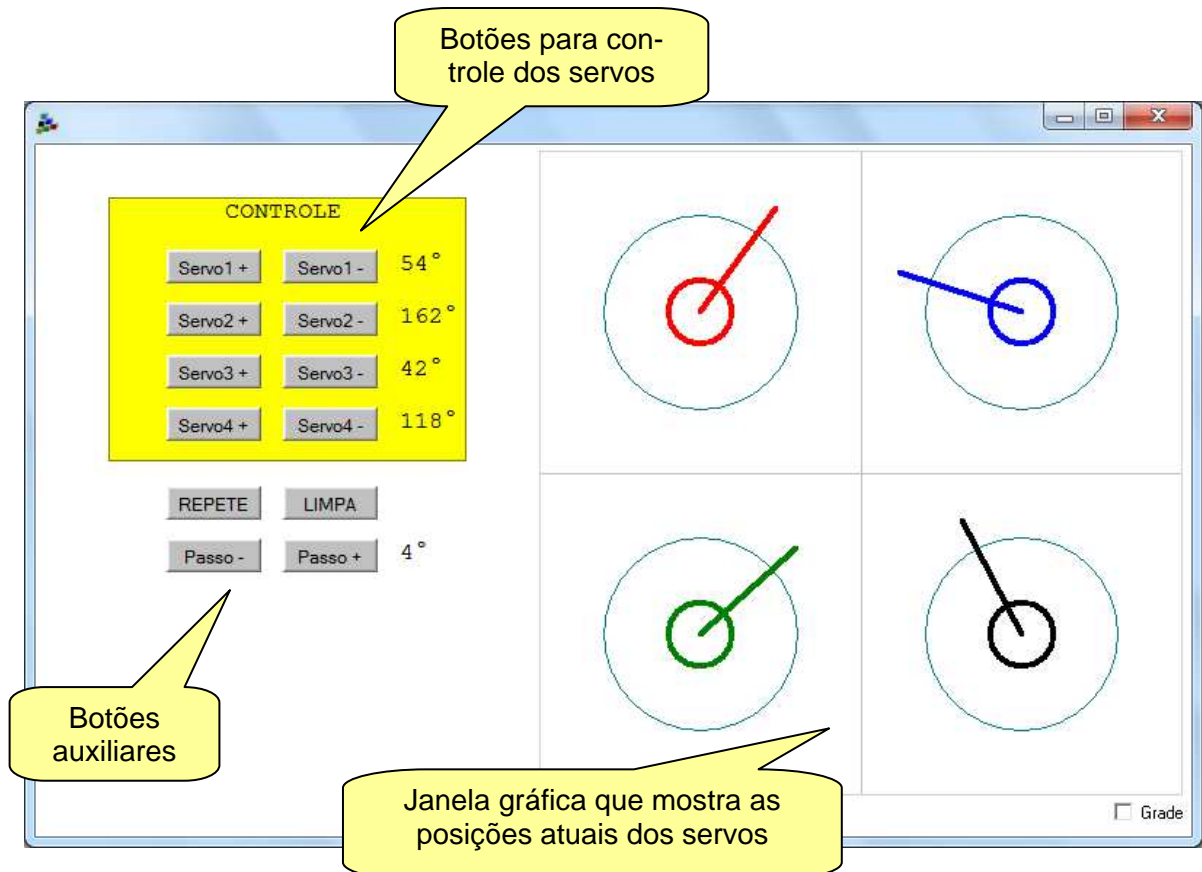
```

rad=a4*pi/180
px=cos(rad)*raio; py=seno(rad)*raio
CorPonto preto
px=px+5;py=py-5
Linha 5,-5/px,py

```

FimSubRotina

A tela de execução deste programa terá o seguinte aspecto:



Os botões auxiliares oferecidos pelo programa têm as seguintes funções:

REPETE: Este botão executa automaticamente todos os comandos realizados pelo usuário até aquele momento.

LIMPA: Promove a limpeza da memória e coloca todos os servos na posição central (90°)

Passo - : Diminui o passo dos motores, ou seja, diminui o valor que será utilizado como incremento ou decremento do ângulo.

Passo + : Aumenta o passo dos motores, ou seja, aumenta o valor que será utilizado como incremento ou decremento do ângulo.

4.6. Projeto 6: Acionamento de um LED RGB (saída analógica)

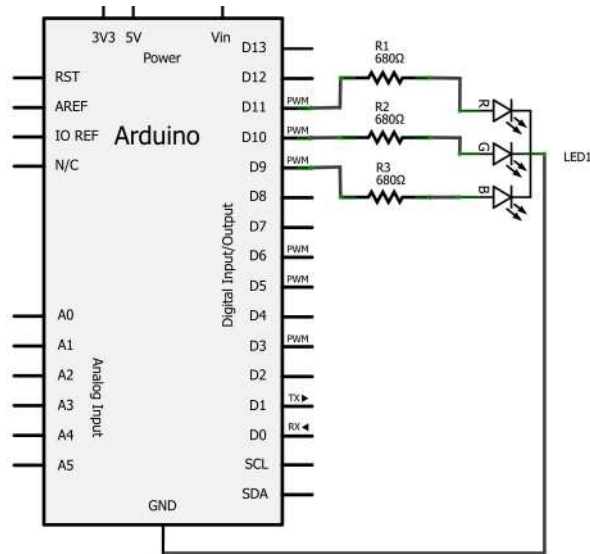
Objetivo:

Demonstrar o uso das saídas analógicas por meio da construção de um programa para controlar as cores de um LED RGB.

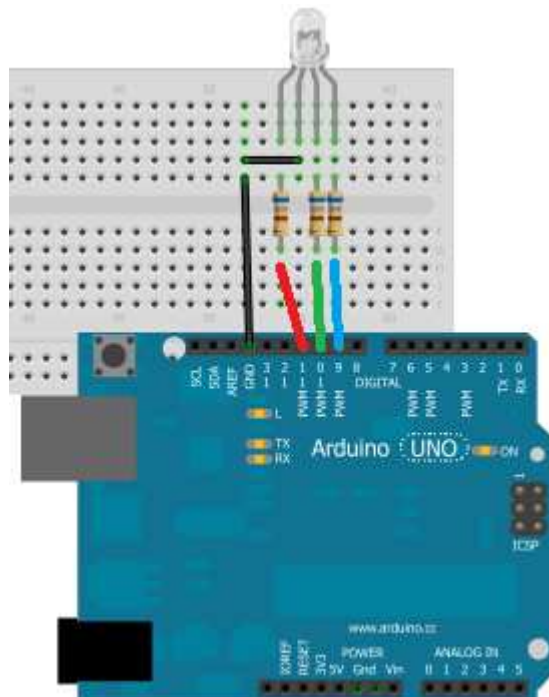
Materiais necessários para a montagem eletrônica:

- 1 LED RGB - catodo comum
- 3 resistores de 680Ω
- 1 placa de prototipagem Arduino
- 1 matriz de contatos

Diagrama esquemático:



Montagem:



Programa INTER-S:

Rotina

```
Declare r, g, b numéricos
ARD_Com 7
[inicio]
Limpa
Receba "Vermelho...: ",r
Receba "Verde.....: ",g
Receba "Azul.....: ",b
ARD_Analog 11,r; Atrase 50
ARD_Analog 10,g; Atrase 50
ARD_Analog 9,b; Atrase 50
VaPara inicio
FimRotina
```

Programa INTER-S comentado:

INSTRUÇÕES	COMENTÁRIOS
Rotina	Início do programa
Declare r, g, b numéricos	Declara as variáveis r , g e b como numéricas. Estas variáveis receberão os valores de intensidade das cores vermelha (red), verde (green) e azul (blue).
ARD_Com 7	Indica o nº da porta serial da placa Arduino (neste caso o nº é 7. Você deve atribuir aqui o nº que o seu sistema gerou para a sua placa Arduino)
[inicio]	Indica uma marca (ponto de retorno)
Limpa	Limpa a tela
Receba "Vermelho...: ",r	Coloca na tela a expressão que está entre aspas e aguarda até que o usuário digite um valor, o qual será atribuído à variável r . O valor digitado deve estar no intervalo de 0 a 255 e corresponde à intensidade da cor vermelha.
Receba "Verde.....: ",g	Coloca na tela a expressão que está entre aspas e aguarda até que o usuário digite um valor, o qual será atribuído à variável g . O valor digitado deve estar no intervalo de 0 a 255 e corresponde à intensidade da cor verde.
Receba "Azul.....: ",b	Coloca na tela a expressão que está entre aspas e aguarda até que o usuário digite um valor, o qual será atribuído à variável b . O valor digitado deve estar no intervalo de 0 a 255 e corresponde à intensidade da cor azul.
ARD_Analog 11,r; Atrase 50	Instrução que envia o valor da variável r (intensidade da cor vermelha) para o pino analógico 11 (pino onde está acoplado o terminal da cor vermelha do LED). Em seguida é dada uma pausa de 50 milissegundos para dar tempo de resposta do Arduino.
ARD_Analog 10,g; Atrase 50	Idem instrução anterior para a cor verde.
ARD_Analog 9,b; Atrase 50	Idem instruções anteriores para a cor azul.
VaPara inicio	Desvia o processamento para o ponto de retorno [inicio]
FimRotina	Finaliza o programa

Exemplo de um programa mais sofisticado para testar a montagem

O programa abaixo, quando executado, apresentará uma tela com 6 botões de comando. Estes botões têm a função de aumentar ou diminuir a intensidade das cores sem que haja a necessidade do usuário digitar os valores:

Rotina

```
Declare br1 BotãoDeComando(4,10,4,13,"-")
Declare bg1 BotãoDeComando(4,14,4,17,"-")
Declare bb1 BotãoDeComando(4,18,4,21,"-")
Declare br2 BotãoDeComando(6,10,6,13,"+")
Declare bg2 BotãoDeComando(6,14,6,17,"+")
Declare bb2 BotãoDeComando(6,18,6,21,"+")
Declare r,g,b,inc numéricos
CorBotão br1,vermelho+
CorBotão br2,vermelho+
CorBotão bg1,verde+
CorBotão bg2,verde+
CorBotão bb1,ciano+
CorBotão bb2,ciano+
r=0;g=0;b=0
inc=5
ARD_Com 7
Processe tela
[inicio]
AtivaBotão
Se br1 = 1 Então
    r=r-inc
    Se r < 0 Então
        r=0
    FimSe
    Processe tela
FimSe
Se br2 = 1 Então
    r=r+inc
    Se r > 255 Então
        r=255
    FimSe
    Processe tela
FimSe
Se bg1 = 1 Então
    g=g-inc
    Se g < 0 Então
        g=0
    FimSe
    Processe tela
FimSe
```

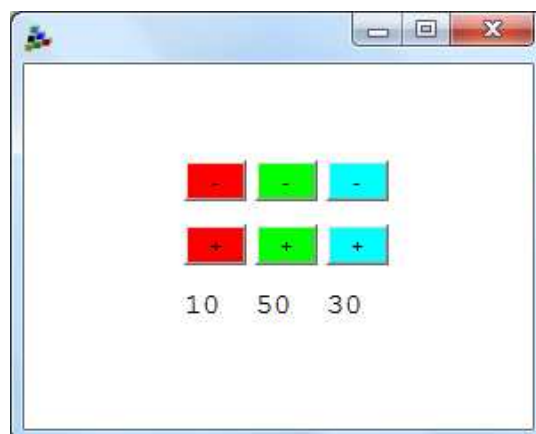
```

Se bg2 = 1 Então
  g=g+inc
  Se g > 255 Então
    g=255
  FimSe
  Processe tela
FimSe
Se bb1 = 1 Então
  b=b-inc
  Se b < 0 Então
    b=0
  FimSe
  Processe tela
FimSe
Se bb2 = 1 Então
  b=b+inc
  Se b > 255 Então
    b=255
  FimSe
  Processe tela
FimSe
VaPara inicio
FimRotina

SubRotina tela
  ARD_Analog 9,r
  ARD_Analog 10,g
  ARD_Analog 11,b
  CorTexto branco
  Moldura 8,1,8,40, branco
  CorTexto preto
  Posicione 8,10; Escreva r
  Posicione 8,14; Escreva g
  Posicione 8,18; Escreva b
FimSubRotina

```

A tela de execução deste programa terá o seguinte aspecto:



Outro exemplo de programa para testar a montagem

Este pequeno exemplo faz com que o LED alterne entre as três cores indefinidamente. Utilizamos a função Ftecla para verificar se a tecla <ESC> (código 27) foi acionada, pois, o programa será finalizado neste momento:

```
Rotina
  Declare tec numérico
  ARD_Com 7
  [inicio]
  tec=Ftecla
  Se tec = 27 Então
    ARD_Analog 11,0; Atrase 50
    ARD_Analog 10,0; Atrase 50
    ARD_Analog 9,0; Atrase 50
    Interrompa
  FimSe
  ARD_Analog 11,255; Atrase 200
  ARD_Analog 11,0; Atrase 50
  ARD_Analog 10,255; Atrase 200
  ARD_Analog 10,0; Atrase 50
  ARD_Analog 9,255; Atrase 200
  ARD_Analog 9,0; Atrase 50
  VaPara inicio
FimRotina
```

4.7. Projeto 7: Reconhecimento de botões (entradas digitais)

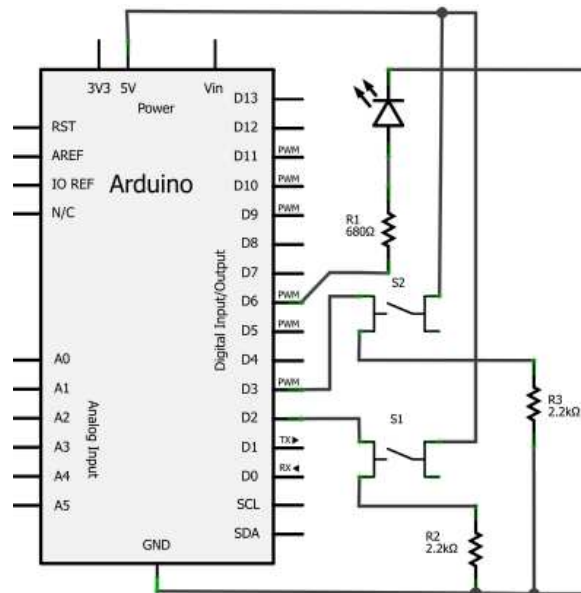
Objetivo:

Demonstrar o uso das entradas digitais da placa Arduino por meio de um programa INTER-S para reconhecer o estado de dois botões acoplados nos pinos digitais 2 e 3.

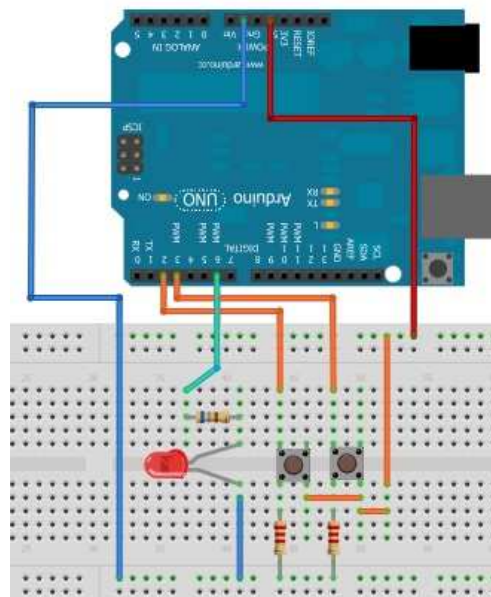
Materiais necessários para a montagem eletrônica:

- 1 LED 5mm (cor de sua preferência)
- 2 resistores de 2,2K Ω
- 1 resistor de 680K Ω
- 2 botões do tipo *pushbutton*
- 1 placa de prototipagem Arduino
- 1 matriz de contatos

Diagrama esquemático:



Montagem:



Programa INTER-S:

Rotina

```
Declare x, flag numérico
ARD_Com 7
flag=0;
ARD_DesligaPino 6
Limpa
Escreva "Desligado"
Moldura 3,2,3,4,vermelho
[volta]
ARD_LePino 2
Se ard_leitura_2 = 1 Então
  Se flag = 0 Então
    ARD_LigaPino 6
    Limpa
    Escreva "Ligado";flag=1
    Moldura 3,2,3,4,vermelho+
  Senão
    ARD_DesligaPino 6
    Limpa
    Escreva "Desligado"; flag=0
    Moldura 3,2,3,4,vermelho
  FimSe
FimSe
ARD_LePino 3
Se ard_leitura_3 = 0 Então
  VaPara volta
FimSe
FimRotina
```

Programa INTER-S comentado:

INSTRUÇÕES	COMENTÁRIOS
Rotina	Início do programa
Declare flag numérica	Declara a variável flag como numérica. Esta variável sinalizará o estado do pino 6 (o pino 6 é a saída digital na qual está acoplado o anodo do LED)
ARD_Com 7	Indica o nº da porta serial da placa Arduino (neste caso o nº é 7. Você deve atribuir aqui o nº que o seu sistema gerou para a sua placa Arduino)
flag=0;	Inicia a variável flag com zero, sinalizando que o LED está inicialmente apagado.
ARD_DesligaPino 6	Leva o pino digital 6 ao nível zero (desliga o pino) fazendo com que o LED efetivamente fique apagado.
Limpa	Limpa a tela.
Escreva "Desligado"	Escreve a palavra "Desligado" na tela.
Moldura 3,2,3,4,vermelho	Desenha um retângulo com preenchimento de cor vermelho escuro, simbolizando o LED apagado.

[volta]	Indica uma marca (ponto de retorno)
ARD_LePino 2	Lê o estado do pino de entrada 2 (onde está acoplado o botão <i>pushbutton</i> 1). Se o botão estiver acionado, a variável de ambiente ard_leitura_2 receberá como conteúdo o valor 1; caso contrário, receberá o valor 0.
Se ard_leitura_2 = 1 Então	Verifica se o valor da variável de ambiente ard_leitura_2 é igual a 1...
Se flag = 0 Então	Caso positivo, verifica se o valor da variável flag for igual a zero (zero=LED apagado)...
ARD_LigaPino 6	... caso positivo, ou seja, se flag=0 , envia comando para ligar o pino 6 (acende o LED)...
Limpa	... limpa a tela ...
Escreva "Ligado";flag=1	... escreve na tela a palavra "Ligado" e ajusta a variável flag para 1...
Moldura 3,2,3,4,vermelho+	...desenha um retângulo com preenchimento de cor vermelho claro, simbolizando o LED aceso.
Senão	... caso contrário, ou seja, se o valor da variável flag for igual a um (um=LED aceso)...
ARD_DesligaPino 6	... envia comando para desligar o pino 6 (apagar o LED)...
Limpa	... limpa a tela ...
Escreva "Desligado"; flag=0	... escreve na tela a palavra "Desligado" e ajusta a variável flag para 0...
Moldura 3,2,3,4,vermelho	Desenha um retângulo com preenchimento de cor vermelho escuro, simbolizando o LED apagado.
FimSe	Finaliza a última estrutura de decisão aberta
FimSe	Finaliza a primeira estrutura de decisão aberta
ARD_LePino 3	Lê o estado do pino de entrada 3 (onde está acoplado o botão <i>pushbutton</i> 2). Se o botão estiver acionado, a variável de ambiente ard_leitura_3 receberá como conteúdo o valor 1; caso contrário, receberá o valor 0.
Se ard_leitura_3 = 0 Então	Verifica se o valor da variável de ambiente ard_leitura_3 é igual a zero...
VaPara volta	... caso positivo, ou seja, se o botão 2 não estiver acionado, o programa retornará ao ponto [volta], repetindo o processo.
FimSe	Finaliza a estrutura de decisão aberta
FimRotina	Finaliza o programa

Notas:

O programa apresentado acima inicialmente mostrará uma tela com a expressão "Desligado" na primeira linha e logo abaixo um retângulo com preenchimento vermelho escuro, o que sinalizará o LED apagado.

Quando o usuário pressionar o botão 1 (aquele acoplado ao pino digital 2 do Arduino) o LED acenderá e na tela surgirá a expressão "Ligado" seguida por um retângulo com preenchimento vermelho claro, indicando o estado do LED (agora aceso).

Pressionando mais uma vez o botão 1, o LED será apagado e o novo estado será exibido na tela. Observe que o botão 1 tem a função de alternar o estado do LED (acendê-lo e apagá-lo).

O botão 2 (aquele acoplado ao pino digital 3 do Arduino) tem uma única função: encerrar o programa.