



(IoT) Solução Fácil e Abrangente com Delphi Android, Beacons e Arduino

Projeto Arduino

O que é Arduino

É uma ferramenta para criar computadores que podem sentir e controlar mais o mundo que seu PC. Ele é uma plataforma física de computação de código aberto baseado numa simples placa microcontroladora, e um ambiente de desenvolvimento para escrever o código para a placa.

Nosso Propósito

Montar o essencial em equipamento (placa e shield), desenvolver e embarcar código suficiente para receber instruções e devolver informações.

Na prática esperamos receber um valor de uma Tag, capturada pelo Shield RFID, e em seguida baseado em instrução enviado pelo software de controle (Servidor DataSnap), acionar algum equipamento controlado pela Placa Arduino.

Para tanto, segue um dois roteiros passo-a-passo, uma para montagem do equipamento e configuração dos elementos a serem acionados.



(IoT) Solução Fácil e Abrangente com Delphi Android, Beacons e Arduino

Montando a Placa Arduino e os Acessórios

Passo 01

Conectar a Placa Arduino UNO a porta USB de seu computador, observando o resultado na Figura 01, onde podemos ver o led verde em estado ON (ligado).

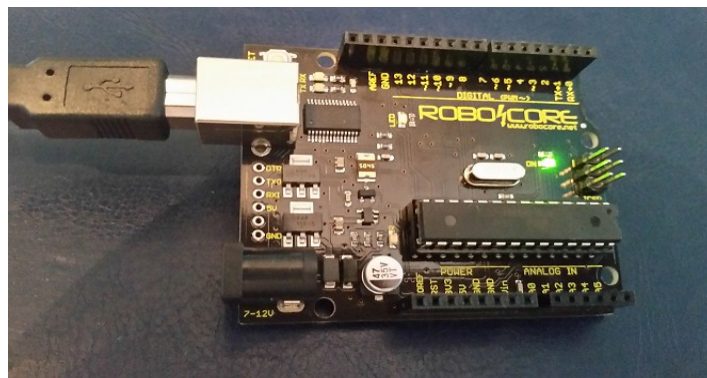


Figura 01

Passo 02

Obtenha um Shield Leitor RFID conforme Figura 02, aqui no exemplo o modelo RFID-RC522A. Também adquira duas ou mais Tags RFID conforme Figura 03, sejam cartões ou chaveiros.

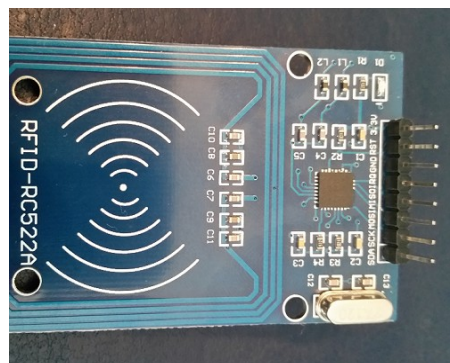


Figura 02



(IoT) Solução Fácil e Abrangente com Delphi Android, Beacons e Arduino

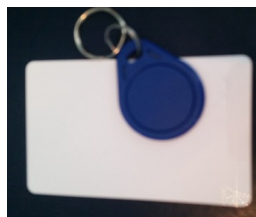


Figura 03

Passo 03

Objetivando conectar o Shield Leitor RFID a Placa Arduino, se organize separando os seguintes itens de seu Kit Arduino, observados na Tabela 01.

Tabela 01		
	Item	Quantidade
	Shield Leitor RFID - RC522A	1
	Cabos Macho/Fêmea	7

Passo 04

Tomando por base a Tabela 02, proceda a conexão dos cabos aos referidos pinos no lado Shield Leitora RFID e Placa Arduino. Observe a Placa e Shild montados (Figura 04) com os devidos pinos conectados.

Tabela 02	
Leitor RFID	Placa Arduino
SDA	Pino 10
SCK	Pino 13
MOSI	Pino 11
MISO	Pino 12
IRQ (não conectar)	
GND	Pino GND
RST	Pino RST
3,3 volts	Pino 3,3 volts



(IoT) Solução Fácil e Abrangente com Delphi Android, Beacons e Arduino

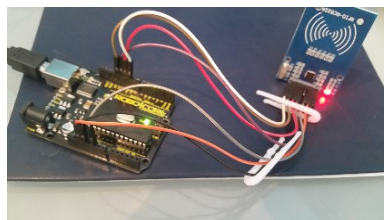


Figura 04

Passo 05

Para que possamos testar todo o ciclo, agora, vamos configurar um Led Verde para acender quando um comando específico for enviado ao software embarcado. A Tabela 03 relaciona os itens mínimos necessário.

Tabela 03

Item	Quantidade
Protobord	1
Cabos Macho/Macho	2
Resistor	1
Led Verde	1

Passo 06

Não temos muito trabalho aqui, bastando seguir as instruções enumeradas na Tabela 04 para que completemos a estrutura física para nosso projeto. A Figura 05 ajudará na identificação dos componentes e em qual posição deve ser acoplado aos furos da Protobord.

Tabela 04

Item	Instrução
Led Verde	Basta espetar na Protobord, naturalmente uma perna em cada furo lado a lado.
Resistor (ponta 1)	Espetar no protobord, no furo a frente a perna - (carga negativa) do Led Verde.
Cabo Macho\Macho Amarelo	Plugar no Pino GND da Placa Arduino. Corrente negativa no caso.
Cabo Macho\Macho Vermelho	Plugar no Pino 8 da Placa Arduino. Fornecerá carga positiva via instrução.



(IoT) Solução Fácil e Abrangente com Delphi Android, Beacons e Arduino

Resistor (ponta 2)

Espetar no proobord, no furo a frente a perna + (carga positiva) do Led Verde.

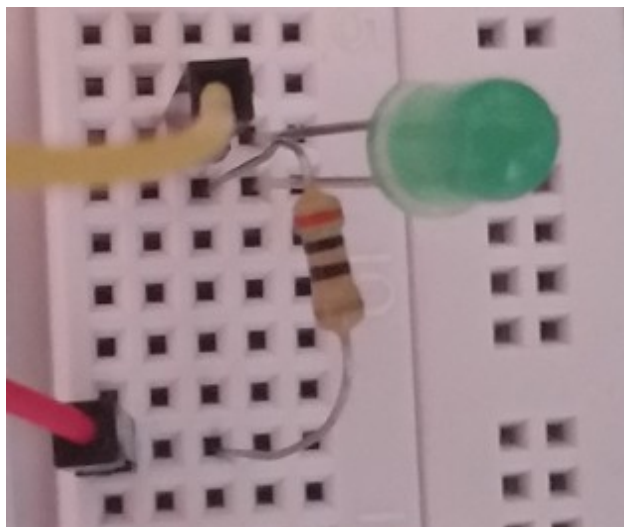


Figura 05



(IoT) Solução Fácil e Abrangente com Delphi Android, Beacons e Arduino

Criando e Embarcando o Software de Controle na Placa Arduino UNO

Passo 01

Instale o software de desenvolvimento do Arduino. Em seguida, conforme Figura 02, configure uma porta autorizada em seu PC ou Note.

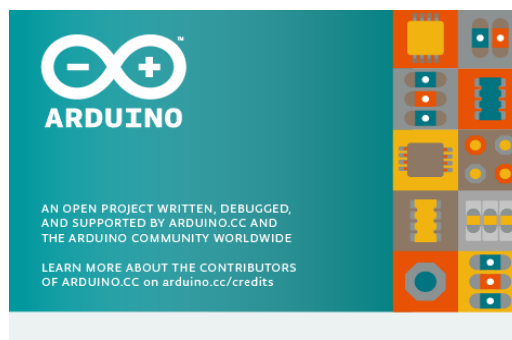


Figura 01

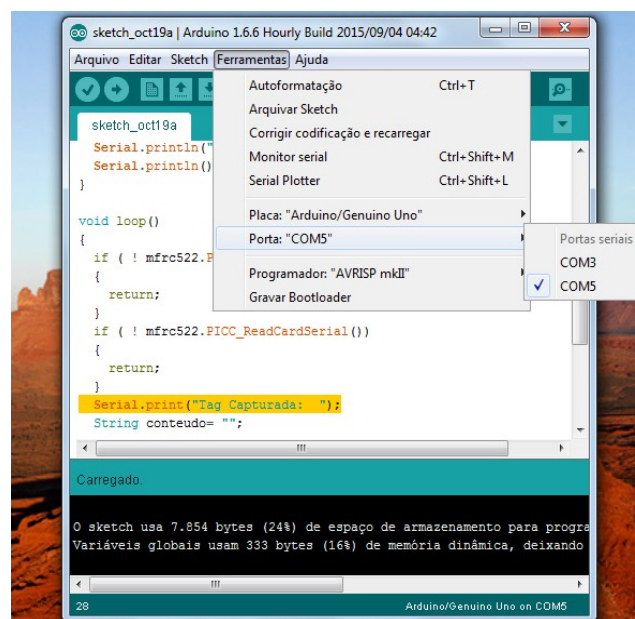


Figura 02



(IoT) Solução Fácil e Abrangente com Delphi Android, Beacons e Arduino

Passo 02

Instalado a ferramenta de desenvolvimento, inicie o ambiente, em seguida, acione menu Arquivo | Novo para criar o projeto inicial. A Figura 03 é o resultado desta ação.

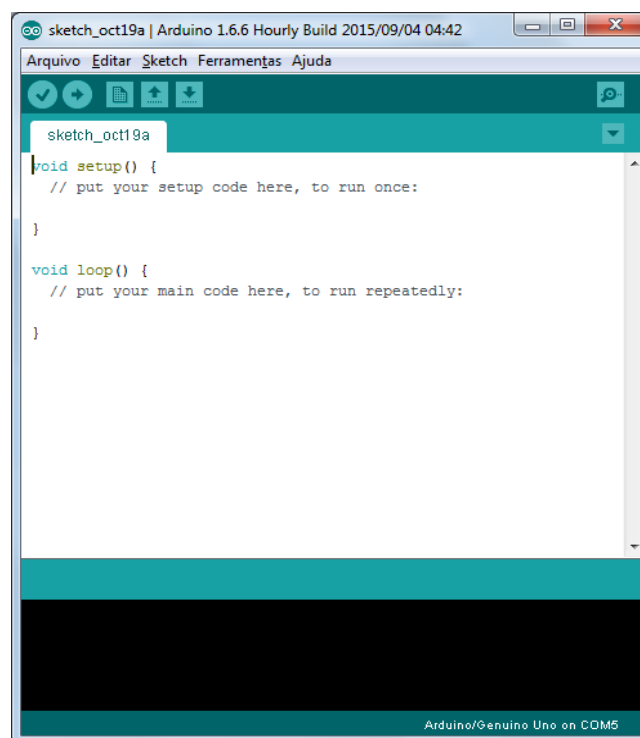


Figura 03

Considerações:

Repare os dois métodos iniciais, `void setup()` e `void loop()`. Respectivamente representam o método executado somente no início (carga) do aplicativo na Placa Arduino, e, o método que será executado de forma contínua, processando as rotinas predefinidas para o aplicativo.

Naturalmente podemos controlar o tempo de ciclo do método `void loop()`, adequando a realidade da tarefa. Também podemos lançar mão do controle por intermédio de um temporizador.



(IoT) Solução Fácil e Abrangente com Delphi Android, Beacons e Arduino

Passo 03

Aplique o código da listagem 01 ao projeto recém criado, substituindo o código pelo da referida listagem.

Listagem 01

```
1.  #include <SPI.h>
2.  #include <MFRC522.h>
3.  #define SS_PIN 10
4.  #define RST_PIN 9
5.  MFRC522 mfrc522(SS_PIN, RST_PIN);
6.
7.  void setup() {
8.    Serial.begin(9600);
9.    SPI.begin();
10.   mfrc522.PCD_Init();
11.   Serial.println("Aproxime o seu cartao do leitor...");
12.   Serial.println();
13. }
14.
15. void loop() {
16.   if ( ! mfrc522.PICC_IsNewCardPresent()) { return; }
17.   if ( ! mfrc522.PICC_ReadCardSerial()) { return; }
18.   Serial.print("Tag Capturada: ");
19.   String conteudo= "";
20.   for (byte i = 0; i < mfrc522.uid.size; i++) {
21.     conteudo.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : ""));
22.     conteudo.concat(String(mfrc522.uid.uidByte[i], HEX));
23.   }
24.   conteudo = "INI111" + conteudo + "222FIN";
25.   conteudo.toUpperCase();
26.   Serial.println(conteudo);
27.   Serial.println("");
28.   delay(500);
29. }
```




(IoT) Solução Fácil e Abrangente com Delphi Android, Beacons e Arduino



(IoT) Solução Fácil e Abrangente com Delphi Android, Beacons e Arduino

Passo 04

Vamos promover o primeiro teste. Para tanto, observe na Figura 02, canto superior esquerdo, um ícone representando uma lupa. Clique nele para exibir o utilitário de leitura serial. Isso mesmo, tal utilitário exibirá todos os prints codificados na rotina da Listagem 01.

Confira na Figura 04 o leitor serial.

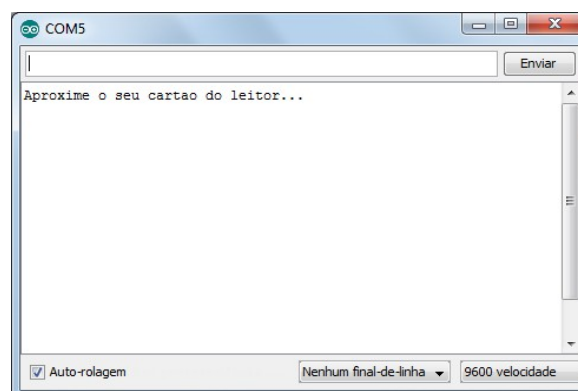


Figura 04

Passo 05

Passa alguma tag e veja o resultado. No meu caso, estou utilizando uma Tag Cartão e uma Tag Chaveiro. Confira na Figura 04 o resultado.

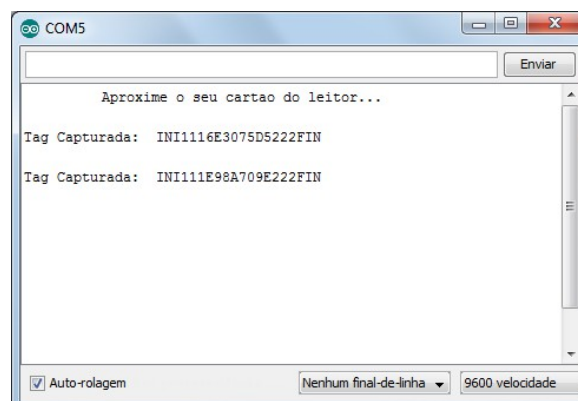


Figura 04



(IoT) Solução Fácil e Abrangente com Delphi Android, Beacons e Arduino

Comentários do Código da Listagem 01

Linha	Comentário
1	Include da biblioteca SPI. Habilita objeto Serial. Menu Sketh Include Library.
2	Include da biblioteca MERCC522. Habilita funções do Shild Leitor. Menu Sketh Include Library.
3	Define SS_PIN como pino 10 da placa.
4	Define RST_PIN como pino 09 da placa.
5	Define objeto do tipo MFRC522 para uso dos métodos de captura do Leitor RFID.
7	Inicia Método void setup(). Método executado exclusivamente na carga do software na placa.
8	Usando objeto Serial, Inicia e define o bound em 9600. Velocidade de leitura do Shield Leitor RFID.
9	Inicia objeto SPI.
10	Inicializa objeto mfrc522.
11	Utiliza o método println(imprime e pula de linha) para enviar uma mensagem para porta serial.
12	Imprime uma linha em branco na porta serial.
15	Inicia Método void loop(). Método executado em loop por tempo indeterminado.
16	Verifica se foi capturado algum valor de Tag pelo Shield Leitor. Se não, aborta o ciclo corrente.
17	Tenta ler conteúdo de Tag capturada, abortando o ciclo corrente do lopp caso não consiga.
18	Imprme na porta serial um texto.
19	Define e inicializa variável do tipo String.
20	Inicia um laço “for”, lendo os bytes para a quantidade de bytes capturados da Tag.
21	Concatena um zero para bytes menor que hexa 10 e vazio para maiores.
22	Concatena o valor Hexa para o byte lido.
24	Concatena “INI111” a esquerda da variável “conteudo” e “222FIN”. Indicando marcação para o tratamento posterior. Isso se dará no aplicativo DataSnap subjacente.
25	Converte o conteudo da variável de mesmo nome em caixa alta.
26	Imprime na porta serial o valor da Tag concatenado com prefixo e sufixo na linha 24.
27	Imprime na serial uma linha em branco
28	Executa o método delay por ½ segundo, objetivando retardar o processo para o próximo ciclo do loop.



(IoT) Solução Fácil e Abrangente com Delphi Android, Beacons e Arduino

Passo 06

Acrescente o código abaixo (Listagem 02) ao método void loop(), ficando este antes da implementação da linha 16 da Listagem 01. Na prática no início do método. Quanto ao método void setup(), acrescente ao final a linha de código da Listagem 03. Por fim, na área de declaração de objetos e variáveis, acrescente as linhas de código da Listagem 04.

Nota:

Tais instruções, faz acender a luz de um Led Verde acoplado a Protobord, respondendo a entrada de instrução “CMDATIVAR” via Porta Serial, oriundo do sistema de controle DataSnap.

Listagem 02

```
1. while (Serial.available()) {  
2.     char c = Serial.read();  
3.     readString += c;  
4. }  
5. if (readString == "CMDATIVAR") {  
6.     digitalWrite(ledPin, HIGH);  
7.     delay(1000);  
8.     digitalWrite(ledPin, LOW);  
9. }  
10. for(int i = 0; i < 10; i++) {  
11.     Serial.read();  
12.     readString = "";  
13. }
```

Listagem 03

```
1. pinMode(ledPin, OUTPUT);
```

Listagem 04

```
1. const int ledPin = 8;  
2. const uint8_t BUFFER_SIZE = 10;  
3. char buffer[BUFFER_SIZE];  
4. String readString;
```




(IoT) Solução Fácil e Abrangente com Delphi Android, Beacons e Arduino

Desenvolvendo o Servidor DataSnap

Passo 01

Partiremos do princípio que todos conhecem o básico para criar um Servidor DataSnap. Se não for o caso, sugiro a leitura do arquivo PDF intitulado “ApostilaDataSnap.pdf”, onde passo-a-passo, instruímos o desenvolvimento de um Servidor DataSnap.

Passo 02

Com o aplicativo DataSnap já criado, desconsideraremos os módulos adicionados pelo processo do wizard, focando exclusivamente no adicionar de um ServerModule para acomodar o acesso aos dados da tabela que armazena o registro de Tags autorizadas a utilizarem o sistema. Como exemplo, Tags cujo o portado será autorizado entrar no empreendimento. Ver Figura 01 para adicionar um ServerModule.

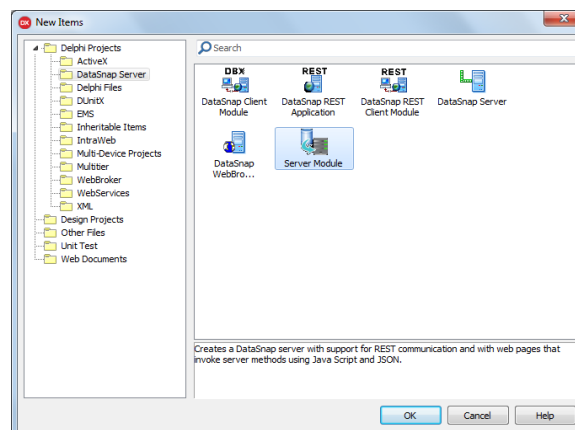


Figura 01

Passo 03

Aplique os componentes de acesso a dados conforme Figura 2. Repare que estamos utilizando middleware de acesso a dados FireDAC além dos controles DataSetProvider e ClientDataSet para manipulação dos dados, a saber, tabela TAG do banco de dados PALESTRA2015

Nota:



(IoT) Solução Fácil e Abrangente com Delphi Android, Beacons e Arduino

Havendo dúvidas de como configurar e aplicar o código necessário a esta etapa, consulte o código fonte junto ao pacote com todo material da Palestra.



(IoT) Solução Fácil e Abrangente com Delphi Android, Beacons e Arduino

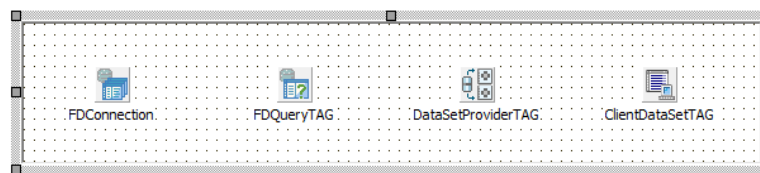


Figura 02

Script da tabela TAG.

```
REATE TABLE [TAG](
    [ID_TAG] [int] NOT NULL,
    [NUMERO] [varchar](50) NOT NULL,
    CONSTRAINT [PK_TAG] PRIMARY KEY CLUSTERED
(
    [ID_TAG] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

Duas Tags Cadastradas.

ID_TAG	NUMERO
10	6E3075D5
20	E98A709E

Passo 04

Na listagem 01, abaixo, implementamos o método que devem ser aplicados a este ServerModule. Proceda aplicando os método na área Public da classe. A listagem 01 exibe o código final.



(IoT) Solução Fácil e Abrangente com Delphi Android, Beacons e Arduino

Listagem 01

```
1.  unit U_DSMPalestra;
2.
3.  interface
4.
5.    Código omitido aqui....
6.
7.  type
8.    TDSServerModule1 = class(TDSServerModule)
9.    Código omitido aqui....
10.
11. public
12.   { Public declarations }
13.   function ExisteTAG(numero : string) : boolean;
14.
15. implementation
16. Código omitido aqui....
17.
18. { TDSServerModule1 }
19.
20. function TDSServerModule1.ExisteTAG(numero: string): boolean;
21. begin
22.   ClientDataSetTAG.Close;
23.   ClientDataSetTAG.Params[0].AsString:=numero;
24.   ClientDataSetTAG.Open;
25.   if ClientDataSetTAG.RecordCount > 0 then
26.     Result:=True
27.   else
28.     Result:=False;
29. end;
30.
31. end.
```




(IoT) Solução Fácil e Abrangente com Delphi Android, Beacons e Arduino

Passo 05

Conforme a Figura 03, adicione um Tmemo ao formulário Form1, nomeando-o como MemoListaTag. Aplique ainda um Ttimer, nomeando-o como TimerLerSerial.

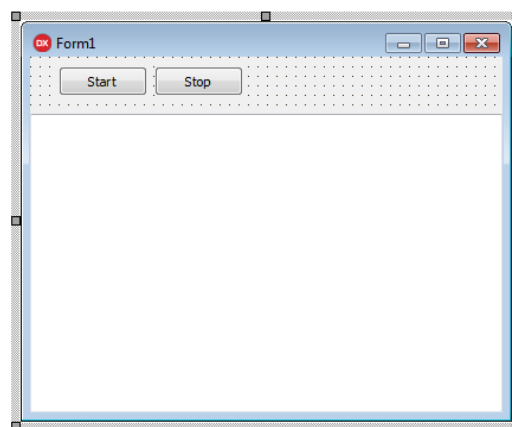


Figura 03

Passo 06

Visando implementar os tipos, constantes e métodos para leitura e escrita via Porta Serial, pela qual receberemos e enviaremos instruções a Placa Arduino, proceda aplicar o código da Listagem 02 a área private da classe Form1.

Listagem 02

1. H : THandle;
2. DSMPalestra : TDSMPalestra;
3. FRespTmp : array[0..1024] of Char; FHandleCom : THANDLE;
4. const
5. dcb_Binary = \$00000001; dcb_ParityCheck = \$00000002;
6. dcb_DtrControlEnable = \$00000010; dcb_OutX = \$00000100;
7. dcb_InX = \$00000200; dcb_RtsControlEnable = \$00001000;
- 8.
9. function AbrePorta(Porta : Pchar) : THANDLE;
10. procedure LerPorta;



(IoT) Solução Fácil e Abrangente com Delphi Android, Beacons e Arduino

11. `procedure EscreveDados(Cmd : string);`



(IoT) Solução Fácil e Abrangente com Delphi Android, Beacons e Arduino

Passo 7

Da mesma forma que no passo anterior, agora obtendo o código da Listagem 03, Listagem 04 e Listagem 05, coloque a implementação dos métodos na classe Form1.

Listagem 03

```
1. function TForm1.AbrePorta(Porta : Pchar) : THANDLE;
2. var
3.   dcbPar : TDCB;
4.   Timeouts : TCOMMTIMEOUTS;
5. begin
6.   SetLastError(0);
7.   Result := CreateFile(Porta, GENERIC_READ or GENERIC_WRITE,
8.     0, nil, OPEN_EXISTING, 0, 0);
9.   if Result = INVALID_HANDLE_VALUE then
10.    Result := 0;
11.   if Result = 0 then
12.    Exit;
13.   try
14.     SetCommMask(Result, 0);
15.     SetupComm(Result, 4096, 0);
16.     FillChar(dcbPar, SizeOf(dcbPar), 0);
17.     dcbPar.DCBlength := SizeOf(dcbPar);
18.     if not GetCommState(Result, dcbPar) then
19.       raise Exception.Create('Erro lendo configurações da porta');
20.     dcbPar.BaudRate := CBR_9600;
21.     dcbPar.ByteSize := 8;
22.     dcbPar.Parity := NOPARITY;
23.     dcbPar.StopBits := ONESTOPBIT;
24.     dcbPar.wReserved := 8;
25.     dcbPar.XonLim := 4096 div 4;
26.     dcbPar.XoffLim := 4096 div 4;
27.     dcbPar.XonChar := #17;
28.     dcbPar.XoffChar := #19;
```



(IoT) Solução Fácil e Abrangente com Delphi Android, Beacons e Arduino

```
29. dcbPar.ErrorChar := #0;
30. dcbPar.Flags := dcb_Binary or dcb_DtrControlEnable;
31. dcbPar.Flags := dcbPar.Flags or dcb_RtsControlEnable;
32. if dcbPar.Parity > 0 then
33.     dcbPar.Flags := dcbPar.Flags or dcb_ParityCheck;
34. if not SetCommState(Result, dcbPar) then
35.     raise Exception.Create('Erro escrevendo configurações da porta');
36. GetCommTimeouts(Result, Timeouts);
37. Timeouts.ReadIntervalTimeout := 2;
38. Timeouts.ReadTotalTimeoutConstant := 150;
39. SetCommTimeouts(Result, Timeouts);
40. except
41.     PurgeComm(Result, PURGE_TXABORT or PURGE_TXCLEAR or PURGE_RXABORT or PURGE_RXCLEAR);
42.     FileClose(Result);
43.     Result := 0;
44. end;
45. end;
```

Listagem 04

```
1. procedure TForm1.LerPorta;
2. var
3.     vl : integer;
4.     vS : String;
5.     i : integer;
6.     Limpa : string;
7. begin
8.     Limpa:='0';
9.     if H = 0 then
10.         H := AbrePorta('COM5');
11.     if H > 0 then
12.         try
13.             vl := 0;
14.             SetLength(vS, 30);
```




(IoT) Solução Fácil e Abrangente com Delphi Android, Beacons e Arduino

```
15.   while vl < 30 do
16.       begin
17.         if FileRead(H, vS[vl + 1], 1) = 1 then
18.             begin
19.                 Inc(vl);
20.                 if (Pos('INI', vS) > 0) AND (Pos('FIN', vS) > 0) then
21.                     begin
22.                         TimerLerSerial.Enabled:=False;
23.                         FileWrite(H, Limpa, 30);
24.                         sleep(100);
25.                         MemoSerial.Lines.Add(Copy(vS, 7, 8));
26.                         if DSMPalestra.ExisteTAG(Copy(vS, 7, 8)) then
27.                             begin
28.                                 EscreveDados('CMDATIVAR');
29.                             end;
30.                         TimerLerSerial.Enabled:=True;
31.                     end;
32.                 end
33.             else
34.                 begin
35.                     Break;
36.                 end;
37.             end;
38.         finally
39.             if MemoSerial.Lines.Count > 500 then
40.                 MemoSerial.Lines.Clear;
41.             end;
42.         end;
```

Listagem 05

```
1.  procedure TFormComunicacao.EscreveDados(Cmd : string);
2.  var
3.      vl : integer;
4.      vS : String;
```



(IoT) Solução Fácil e Abrangente com Delphi Android, Beacons e Arduino

```
5.  Limpa:string;
6.  begin
7.    if H = 0 then
8.      H := AbrePorta('COM5');
9.    SetLength(Limpa, 1);
10.   Limpa:='0';
11.   if H > 0 then
12.     try
13.       vl := 0;
14.       SetLength(Cmd, 9);
15.       while vl < 9 do
16.         begin
17.           if FileWrite(H, Cmd[vl + 1], 1) = 1 then
18.             begin
19.               Inc(vl);
20.             end;
21.           end;
22.           Sleep(100);
23.           FileWrite(H, Limpa, 1);
24.         finally
25.
26.         end;
27.       end;
```

Passo 08

Finalizando, aplique o código destacado em negrito, respectivamente ao evento Show do Form1 e ao evento Timer do componente TimerLerSerial.

```
procedure TForm1.FormShow(Sender: TObject);
begin
  H := AbrePorta('COM5');
  Sleep(500);
  TimerLerSerial.Enabled:=True;
  DSMPalestra:=TDSMPalestra.Create(Self);
```



(IoT) Solução Fácil e Abrangente com Delphi Android, Beacons e Arduino

end;

```
procedure TForm1.TimerLerSerialTimer(Sender: TObject);  
begin  
    LerPorta;  
end;
```



(IoT) Solução Fácil e Abrangente com Delphi Android, Beacons e Arduino

Sistemas Físico e Lógico em Funcionamento

