Apresentação

A proposta desta apresentação na Embarcadero Conference 2017, vem de encontro a uma corrente denominda IoT (Internet of Things), ou se preferir, em português claro, Internet das Coisas.

Esse movimento assume a necessidade de sistemas cada vez mais inteligentes, poderem interagir com dispositivos das mais variadas naturezas.

Entendamos por dispositivos, sensores como medidor de temperatura, humidade, luminosidade e corpos em movimento. Acrescente a isso Leitores de Tags RFID, seja de aproximação para acesso a um local ou evento, e ainda Leitores de Tags RFID a distância, detectando a presença de um veículo, procedendo assim a permissão de acesso ou não.

Repare que basicamente nos concentramos até aqui em monitoramento para fins de acesso e conforto doméstico. Mas isso não acaba aqui. Pense também na possibilidade de controle de processos industriais, maquinário em geral, fluxo de veículos urbano e tudo, absolutamente tudo que seja interessante ser controlado.

Lançando mão de algumas controladoras, conectando tais dispositivos as mesmas, podemos facilmente capturar, por exemplo, a variação de luminosidade de um ambiente, comandando o aumento de luminosidade de uma lâmpada.

Ecesse cenário é clássico, se aplica a este exemplo e outros, envolvendo tecnológicamente um controladora (Arduino por Exemplo), o sensor de luminosidade e algum código embarcado nesta controladora. Mas, isso naturalmente, em mundo perfeito, controlado, e não no caus no universo IoT.

Considerando que a variaedade e quantidade de dispositivos que devem ser controlados, suas informações coletadas em períodos curtos, sensíveis a presença de pessoas ou veículos, demandando decisão que permite comandar tais controladaras para abrirem portas, cataracas, portões de acesso ou acionar aumento de iluminação, é inevitável a necessidade de lançarmos mão de software externo as controladaras coordenando esse trabalho. Por isso o tema IoT combinado com tecnologia ataSnap do Delphi.

Porque Integrar Delphi e Arduino?

A proposta de unir programão Delphi com prototipagem Arduino e Shields, além de implementar lógica na linguagem do Arduino em um único treinamento, vem da observação e prática de minha atividade diária.

Desenvolvo diariamente para esta plataforma, combinando controladoras Arduino com alguns dispotivos de controle de acesso principalmente.

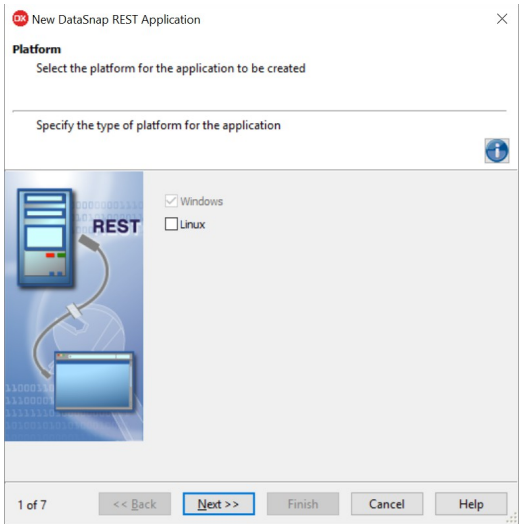
Ainda que a solução para identificação de pessoas e o uso de Tags das mais variadas, seja por intermédio dos leitores (shields) conectados as controladoras Arduino, todo tratamento de regras de negócio e outras mecanismo de validação são operados por um aplicativo Dephi que interage em tempo real com um banco de dados.

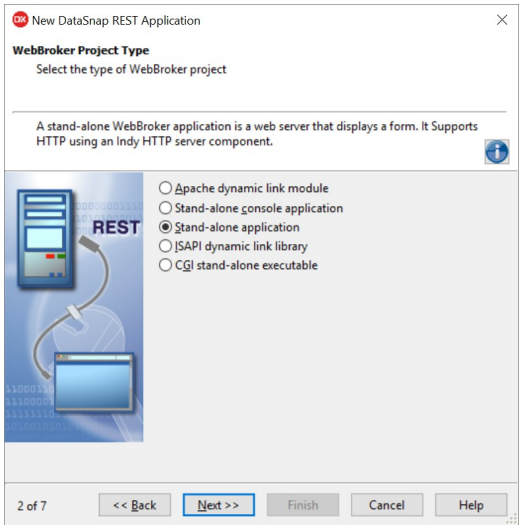
Vale ressaltar, a importância das tecnologias envolvidas, periféricas, tanto ao Arduino com hadware e o Arduino na condição software. A performance nesse tipo de aplicativo, numa visão do todo, requer muita agilidade na decisão, por exemplo, de uma aproximação de um veículo que portando em seu parabrisa uma Tag Veicular, necessita de um tempo médio de 1 segundo para que o sistema num todo, hardware Arduino, software Arduino, combinado com o processo de validação do aplicativo Delphi, que busca tal validação em uma base de dados, seja processado e temo como resposta a baertura de um portão ou cancela de garagem. Na vverdade, 5 segundo é uma eternidade para um processo típico. Em alguns casos, a tolerância é 1 segundo. Assim, justifica-se tanta preocupação com o todo, delegando a cada parte envolvido no processo, tarefas exclusivas dessa camada.

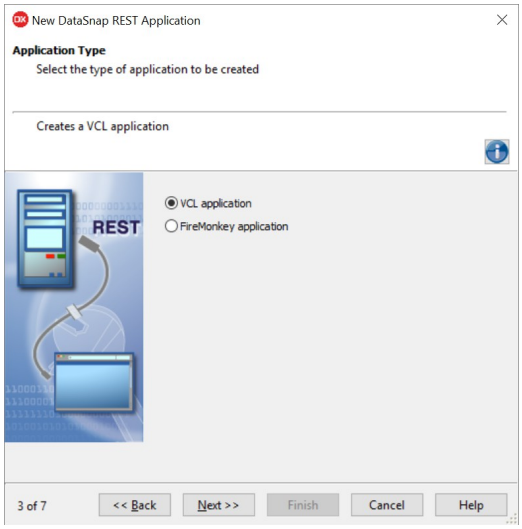
Iniciando o Aplicativo DataSnap

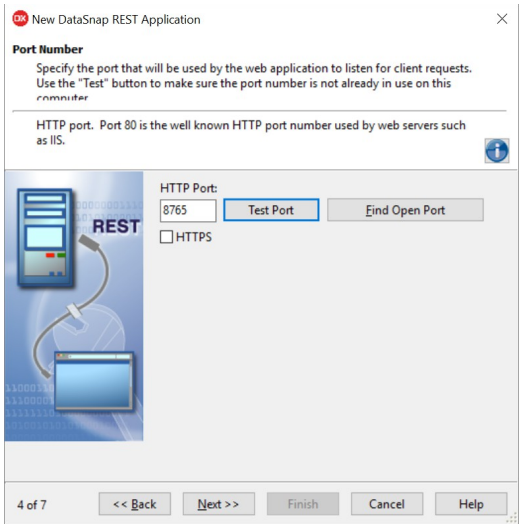
Passo 01

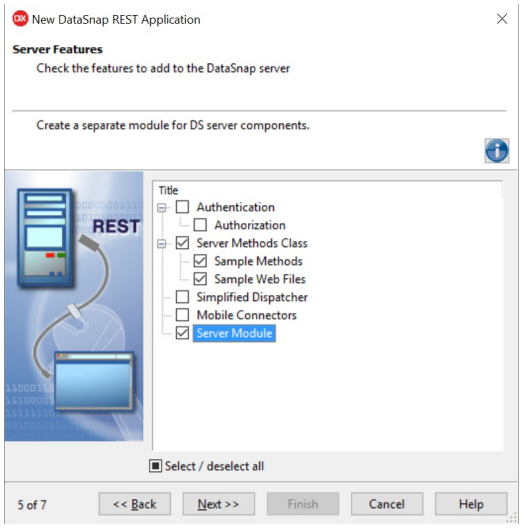
Com a IDE do Delphi, crie um aplicativo Delphi a partir do menu File | New | Other | DataSnap Server, selecionando DataSnap REST Application. Siga os passos representados pela imagens abaixo encerrando esta parte.

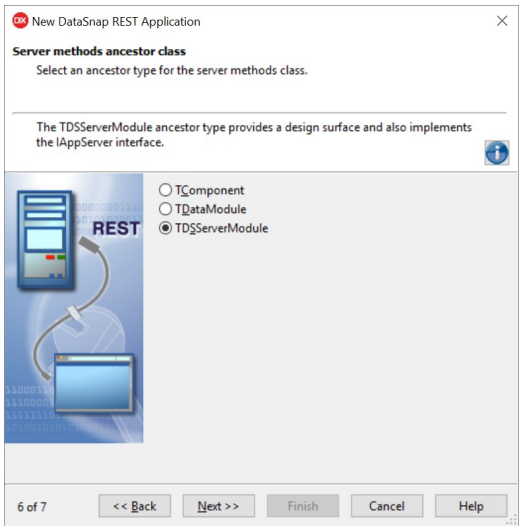





New DataSnap REST Application
Application Type
Select the type of application to be created
Creates a VCL application
REST
VCL application
FireMonkey application
3 of 7
<< Back
Next >>
Finish
Cancel
Help

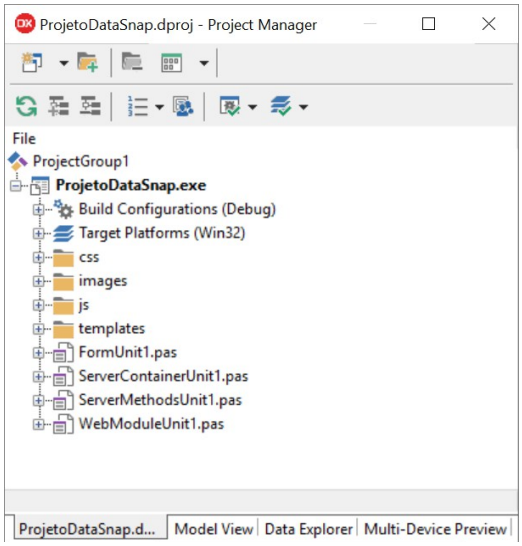

New DataSnap REST Application
Port Number
Specify the port that will be used by the web application to listen for client requests. Use the "Test" button to make sure the port number is not already in use on this computer
HTTP port. Port 80 is the well known HTTP port number used by web servers such as IIS.
REST
HTTP Port:
8765
Test Port
Find Open Port
HTTPS
4 of 7
<< Back
Next >>
Finish
Cancel
Help


New DataSnap REST Application
Server Features
Check the features to add to the DataSnap server
Create a separate module for DS server components.
REST
Title
Authentication
Authorization
Server Methods Class
Sample Methods
Sample Web Files
Simplified Dispatcher
Mobile Connectors
Server Module
Select / deselect all
5 of 7
<< Back
Next >>
Finish
Cancel
Help



Passo 02

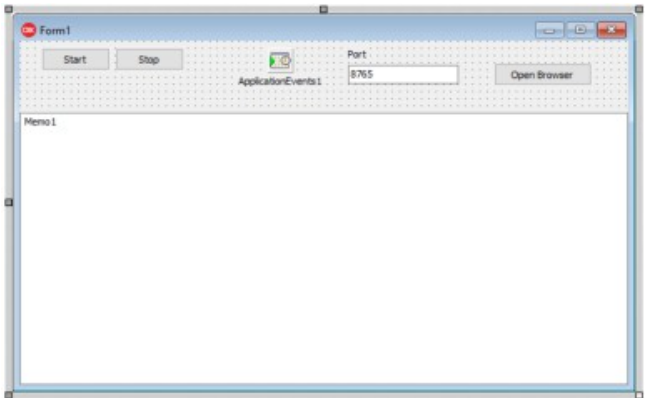
Confira em Project Manager, a estrutura básica do aplicativo gerado.



Passo 03

Altere o layout do FormUnit1 para que fique semelhante a figura que segue. Repare, deve adicionar um objeto Tmemo e alinha na parte inferior do Form.

Este componente Tmemo servirá como exibidor dos processos enviados pelas controladoras ao aplicativo DataSnap.



Passo 04

Adicione ao FormUnit1 um contole TidUDPServer (veja imagem abaixo para melhor identificar). Na sequência, iremos manipular em código algumas propriedades deste componente, visando configurá-lo para interceptar mensagens enviadas pelas controladoras (ou outros processos UDP) a ele.

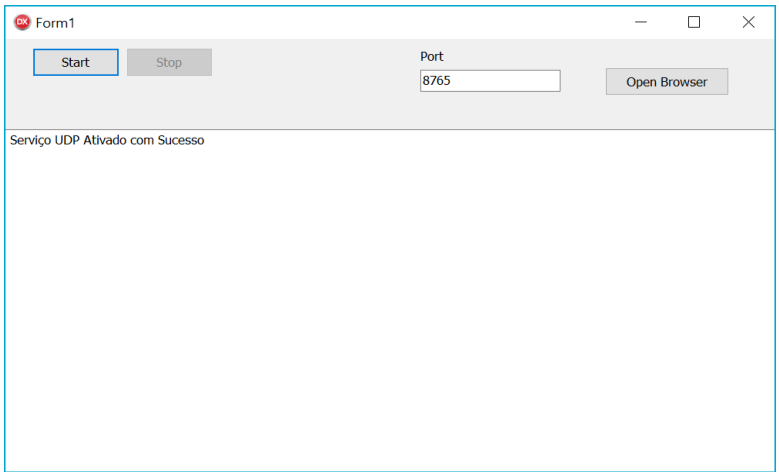
Passo 05

Implemente o código que segue, criando um procedimento para o evento onShow do formulário FormUnit1.

```
procedure TForm1.FormShow(Sender: TObject);
begin
  IdUDPServer1.Bindings.Add.IP:='192.168.25.50';
  IdUDPServer1.Bindings.Add.Port:=26700;
  IdUDPServer1.Active:=True;
  if IdUDPServer1.Active then
    begin
      Memo1.Lines.Add('Serviço UDP Ativado com Sucesso');
    end
  else
    begin
      Memo1.Lines.Add('Serviço UDP Não Ativado');
    end;
end;
```

Repare que estamos definindo IP e Porta que o Serviço UDP está sendo executado. Todo processo, seja de uma controladora Arduino, ou mesmo de um outro aplicativo na mesma rede, enviados para esse endereço IP/Porta será processado conforme cógo no próximo passo. Este mesmo controle, uma vez configurado e ativado, funciona como receptor e transmissor de mensagens entre outros processos UDP.

Repare na imagem abaixo, o projeto em execução.



Passo 06

Agora vamos implementar o código que interceptará em nosso UDP Server, mensagens enviadas por processos UDP Client, sejam de controladoras conectadas a rede ou mesmo processos UDP de outros aplicativos de rede.

Primeiro implemente essa função que vai apoiar nosso código receptor de mensagens.

A função HexToString converte valores hexadecimal para o formato string.

```
function TForm1.HexToString(H: String): String;
Var I : Integer;
begin
  Result:= '';
  for I := 1 to length (H) div 2 do
     Result:= Result+Char(StrToInt('$'+Copy(H,(I-1)*2+1,2)));
end;
```

Ainda neste passo, implemte o código que segue, aplicado ao evento onUDPRead do componente IdUDPServer.

```
procedure TForm1.IdUDPServer1UDPRead(AThread: TIdUDPListenerThread;
  const AData: TIdBytes; ABinding: TIdSocketHandle);
var
  StringRecebida : String;
  i : integer;
begin
   StringRecebida:='';
   for i:=0 to Length(AData) - 1 do
      begin
        StringRecebida:=StringRecebida +
                    HexToString( IntToHex(Integer(AData[i]), 2) );
      end;

   if StringRecebida <> '' then
      begin
        Memo1.Lines.Add('');
        Memo1.Lines.Add('Recebido: ' + StringRecebida + '  -  ' +
                    'IP Origem: ' + ABinding.PeerIP + '  -  ' +
                    'Porta Origem: ' + ABinding.PeerPort.ToString
                   );
      end;
end;
```

Merece destaque o parâmetro Athred, do tipo TidUDPListner, caracterisando o processo em Therad, importante, já que em um ambiente real, vários processos distintos são enviados de vários destinos igualmente distintos. Não haverá epera para execução de procedimentos que forem enviados.

Outro elemento importante, é o parâmetro Adata, do tipo TidBytes, que contem a mensagem enviada de forma serializada. O código que segue transforma essa mensagem em string.

```
StringRecebida:='';
   for i:=0 to Length(AData) - 1 do
      begin
        StringRecebida:=StringRecebida +
                    HexToString( IntToHex(Integer(AData[i]), 2) );
      end;
```

Por fim, não menos importante, temos o parâmetro Abinding, tipado como TidSocketHandle. Utilizamos suas propriedade para recuperar por exemplo o IP e Porta do processo UDP Client, abrindo assim para em tempo real, identificar e comandar por exemplo a Controladora que enviou a mensagem.

```
if StringRecebida <> '' then
    begin
      Memo1.Lines.Add('');
      Memo1.Lines.Add('Recebido: ' + StringRecebida + '  -  ' +
                  'IP Origem: ' + ABinding.PeerIP + '  -  ' +
                  'Porta Origem: ' + ABinding.PeerPort.ToString
                 );
    end;
```

Iniciando o Aplicativo Embarcado na Controladora Arduino Comunicação Bluetooth

Nota:
Baixe gratuitamente a IDE de desenvolvimento para Arduino no site www.arduino.cc, em seguida, instale e pronto.

Este aplicativo fará a comunicação com o aplicativo DataSnap utilizando a Bluetooth.

Tabela de Itens Utilizados

QTD	Item / Descrição
1	Controladora Arduino Mega c/Cabo USB
1	Módulo Bluetooth HC-05
1	Protobord
1	Resistor 20k
1	Resistor 10k
4	Jumpers Macho X Macho
5	Jumpers Macho X Fêmea

Nota: Na prática, qualquer combinação de resistores que utilizado em sequência, atendam ao Divisor de Tensão.

Passo 01

Inicie um novo projeto, para tanto, acesso o menu Arquivo | Novo. Repare que um Sktech (unidade do programa) é criado, com dois métodos básicos, a saber, void setup() e void loop().

Passo 02

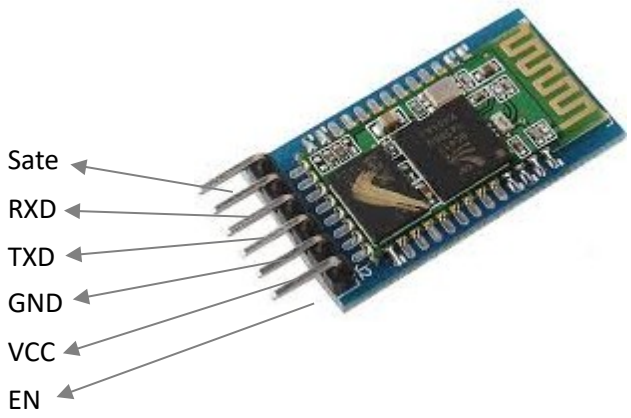
Bom, aplique o código abaixo tanto para Setup quanto Loop. Este código providencia a configuração do módulo para poder ser pareado.

```
void setup(){
  Serial.begin(9600);
  Serial1.begin(38400);
  Serial.println("Iniciado Bluetooth HC-05 -  Comandos AT");
  delay(1000);
  Serial.println("");
}

void loop() {
  while (Serial1.available()){
    Serial.write(Serial1.read());
  }
  delay(50);
  while (Serial.available()){
    Serial1.write(Serial.read());
  }
  delay(100);
}
```

Passo 03

Faça a montagem do Módulo Bluetooth HC-05 na protobord observando a tabela e imagem que segue.



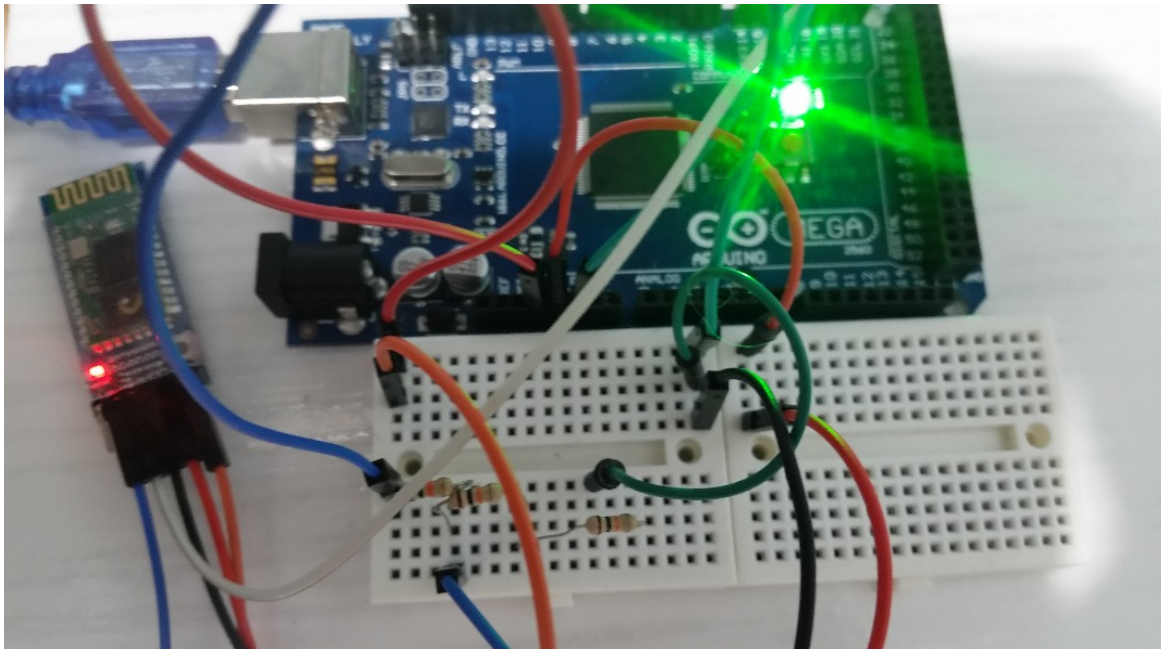
Módulo Bluetooth HC-05	Placa Arduino Mega
State	(não conectar)
RX	Pino 18 (TX1) – com divisor de tensão para 3,3V
TX	Pino 19 (RX1)
GND	GND
VCC - (5V)	5V
EN - (3,3V)	3,3V

Passo 04

Usando o Monitor Serial da IDE do Arduino, execute os comandos ATs listados na tabela abaixo. Procedendo desta forma, entre outras coisas, configuramos o módo, nome do módulo e password para pareamento entre outros dispositivos.

Principais Comandos AT	Definição
AT+ORGL	Reseta o módulo para a configuração padrão.
AT+RMAAD	Remove dispositivos anteriormente pareados.
AT+ROLE=1	Define o modo de operação do módulo como MASTER.
AT+RESET	Reset do módulo após a definição do modo de operação.
AT+CMODE=1	Permite a conexão a qualquer endereço.
AT+INQM=0,5,10	Modo de varredura: padrão, procura por 5 dispositivos ou para a varredura após 10s.
AT+PSWD=1234	Define a senha do módulo mestre, que deve ser a mesma do módulo slave/escravo.
AT+INIT	Inicializa o perfil para transmissão/recepção.
AT+INQ	Inicializa a varredura por dispositivos bluetooth.
AT+NAME=CONTROLADORA_HC05_200	Inicializa a varredura por dispositivos bluetooth.

Confira a imagem que segue para ver a montagem do Módulo HC-05 e a Controladora Arduino Mega, já em funcionamento.



Passo 05

Carrega a Controladora Arduino Mega com o Sketch (programa) listado abaixo. Usando a Serial1 do Arduino Mega, o sistema recebe daods enviados por algum computador ou dispositivo pareado, retornando dados para o mesmo. Na sequência vamos implementar esse processo na aplicação DataSnap, mas poderíamos aplicar litaralmente o mesmo código, usando Delphi é claro, a uma aplicação Mobile Android.

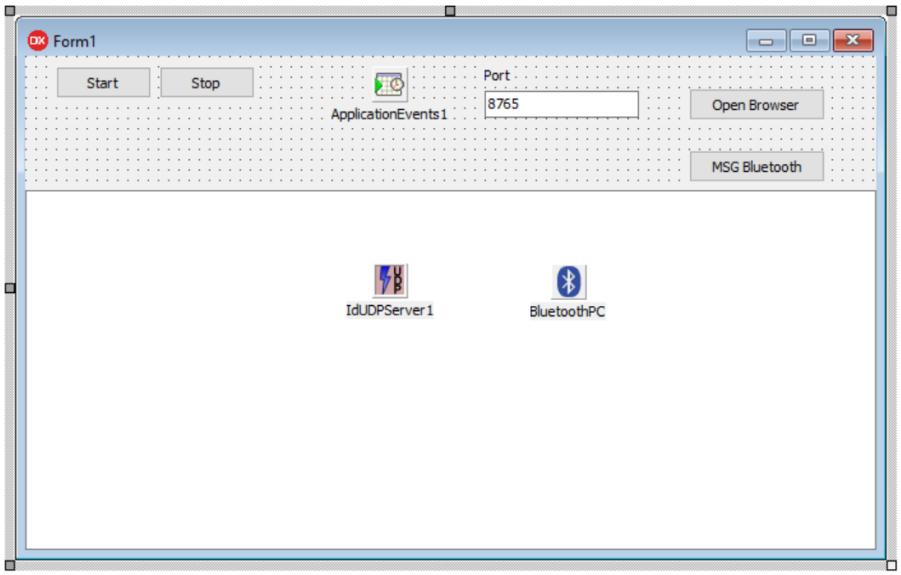
```
int incomingByte;

void setup() {
 Serial.begin(9600);
 Serial1.begin(38400);
 Serial.println("Bluetooth em funcionamento para comunicação");
 Serial.println("");
}

void loop() {
 if (Serial1.available()) {
  incomingByte = Serial1.read();
  if (incomingByte == 'L') {
   Serial1.write("Recebido Letra L");
   Serial.println("Recebido Comando L");
   delay(300);
   incomingByte=0;
  }
  if (incomingByte == 'D') {
   Serial1.write("Recebido Letra D");
   Serial.println("Recebido Comando D");
   delay(100);
   incomingByte=0;
  }
 }
 delay(100);
}
```

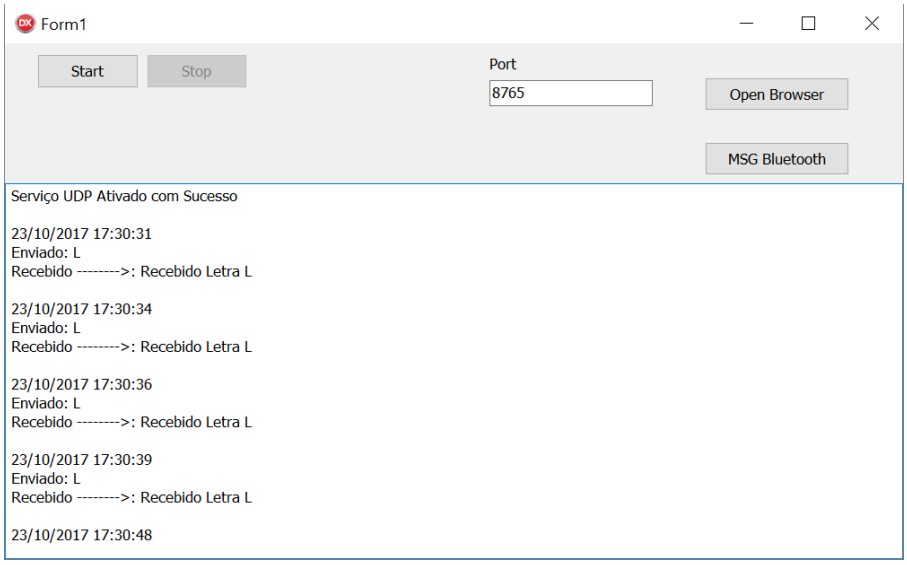
Passo 06

Coloque um componente Bluetooth e um BitBtn no aplicativo DataSnap, conforme imagem do form que segue.



Passo 07

Atualize o código fonte do aplicativo DataSnap para que ele posso fazer a comunicação com a Controladora Arduino Mega, via Módulo Bluetoth HC-05, enviando e recebendo dados. Observe ainda a aplicação DataSnap, na condição de um monitor exibindo essa comunicação.



```
    … Código omitido aqui.
    private
      FServer: TIdHTTPWebBrokerBridge;
      ////////////////////Bluetooth/////////////////////////////////////////////
      FSocket : TBluetoothSocket;
      const UUID = '{00001101-0000-1000-8000-00805F9B34FB}';
      function ObterDevicePeloNome(NomeDevice: String): TBluetoothDevice;
      function ConectarControladoraBluetooth(NomeDevice: String): boolean;
      procedure StartServer;
      function HexToString(H: String): String;
    … Código omitido aqui.
implementation


{$R *.dfm}


uses

  WinApi.Windows, Winapi.ShellApi, Datasnap.DSSession;

    … Código omitido aqui.
    function TForm1.ObterDevicePeloNome(NomeDevice: String): TBluetoothDevice;
    var
      lDevice: TBluetoothDevice;
    begin
      Result := nil;
      for lDevice in BluetoothPC.PairedDevices do
        begin
          if Trim(lDevice.DeviceName) = NomeDevice then
            begin
              Result := lDevice;
            end;
        end;
    end;
```

```
function TForm1.ConectarControladoraBluetooth(NomeDevice: String): boolean;
var
  lDevice: TBluetoothDevice;
begin
  Result := False;
  lDevice := ObterDevicePeloNome(NomeDevice);
  if lDevice <> nil then
  begin
    FSocket := lDevice.CreateClientSocket(StringToGUID(UUID), False);
    if FSocket <> nil then
    begin
      FSocket.Connect;
      Result := FSocket.Connected
    end;
  end;
end;

procedure TForm1.BitBtnMsgBluetoothClick(Sender: TObject);
Var
  strRecebida : string;
begin
  BluetoothPC.Enabled:=True;
  if (FSocket = nil) or (not FSocket.Connected) then
     ConectarControladoraBluetooth('CONTROLADORA_HC05_200');
   if (FSocket <> nil) and (FSocket.Connected) then
      begin
        Memo1.Lines.Add('');
        Memo1.Lines.Add(DateTimeToStr(Now));
        strRecebida:='';
        FSocket.SendData(TEncoding.UTF8.GetBytes('L'));
        Memo1.Lines.Add('Enviado: L');
        sleep(300);
        strRecebida:=Trim(TEncoding.ANSI.GetString(FSocket.ReceiveData));
        Memo1.Lines.Add('Recebido -------->: ' + strRecebida);
        Application.ProcessMessages;
        FreeAndNil(FSocket);
      end;

end;
```

… Código omitido aqui.

Iniciando o Aplicativo Embarcado na Controladora Arduino Comunicação WIFI

Nota:
Baixe gratuitamente a IDE de desenvolvimento para Arduino no site www.arduino.cc, em seguida, instale e pronto.

Este aplicativo fará a comunicação com o aplicativo DataSnap utilizando a rede WIFi. Logo, o código que segue configura o módulo ESP8266, utilizado nesnte exemplo. Confira a tabela abaixo para conferir os itens necessário.

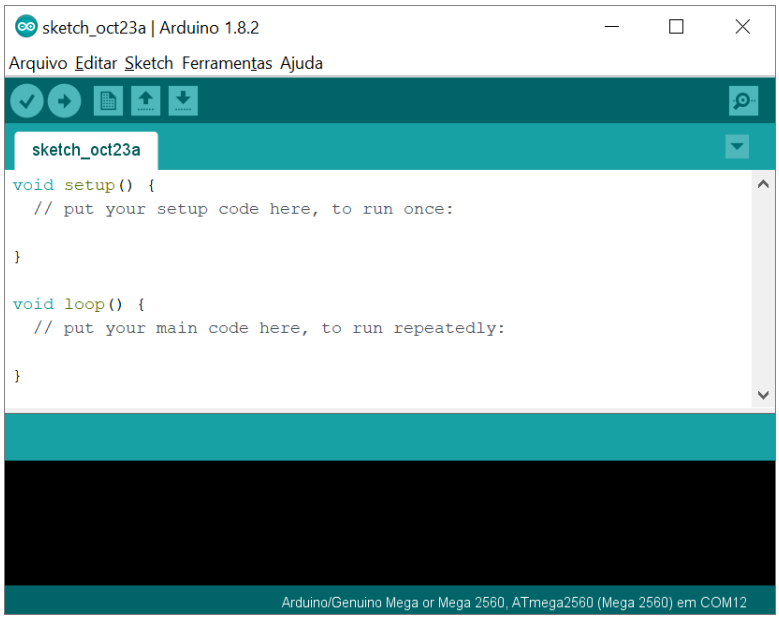
Tabela de Itens Utilizados

QTD	Item / Descrição
1	Controladora Arduino Mega c/Cabo USB
1	Módulo ESP8266
1	Protobord
1	Resistor 20k
1	Resistor 10k
3	Jumpers Macho X Macho
5	Jumpers Macho X Fêmea

Nota: Na prática, qualquer combinação de resistores que utilizado em sequência, atendam ao Divisor de Tensão.

Passo 01

Inicie um novo projeto, para tanto, acesso o menu Arquivo | Novo. Repare que um Sktech (unidade do programa) é criado, com dois métodos básicos, a saber, void setup() e void loop().



Passo 02

Bom, aplique o código abaixo tanto para Setup quanto Loop. Este código providencia a configuração do módulo para um IP e Porta na Controladora, e acessar um IP e Porta no serviço UDP na DataSnap.

```
String IPLIVRE="192.168.25.200";
String MASCARA="255.255.255.0";
String ROTEADOR="192.168.25.1";
String NOMEREDE="GVT-657D";
String SENHAREDE="2803000070";
String IPSERVIDORUDP="192.168.25.50";
String PORTASERVIDORUDP="26700";
String TIPOCONECAOUDP="UDP";
String PORTACONTROLADORAUDP="26700";
int CMDSerial = 0;

void executaCMD(String cmd) {
   Serial2.println(cmd);
}

void setup()
{
    Serial2.begin(115200);
    Serial.begin(115200);
    Serial.println("Digite valores de 1 a 10 para configurar a  WSP8266 ...");
    Serial.println("");
}

void loop()
{
  while (Serial.available() > 0) {
     CMDSerial = Serial.parseInt();

     if (CMDSerial == 1) {
       Serial.println("");
       Serial.println("Comando AT");
       Serial.println("");
       executaCMD("AT");
     }
     if (CMDSerial == 2) {
       Serial.println("");
       Serial.println("Comando AT+CIPSTA_DEF");
```

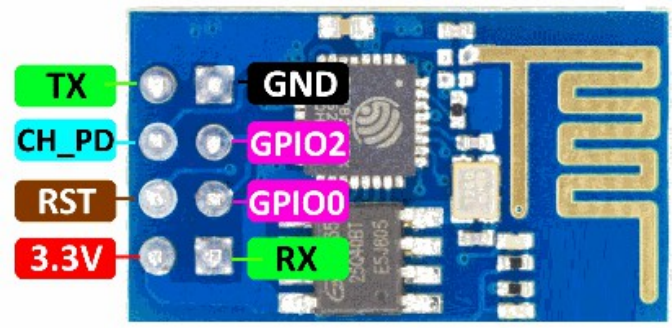
```
  Serial.println("");
  String cmdMudarIP="AT+CIPSTA_DEF=\"" + IPLIVRE + "\",\"" + ROTEADOR +"\",\"" +
   MASCARA + "\"";
  executaCMD(cmdMudarIP);
}
if (CMDSerial == 3) {
  Serial.println("");
  Serial.println("Comaando AT+CWMODE");
  Serial.println("");
  executaCMD("AT+CWMODE=3");
}
if (CMDSerial == 4) {
  Serial.println("Comaando AT+CWJAP");
  Serial.println("");
  String ConexaoRede="AT+CWJAP=\"" + NOMEREDE + "\",\"" + SENHAREDE + "\"";
  executaCMD(ConexaoRede);
}
if (CMDSerial == 5) {
  Serial.println("Comando AT+CIFSR");
  Serial.println("");
  delay(2000);
  executaCMD("AT+CIFSR");
}
if (CMDSerial == 6) {
  Serial.println("Comando AT+CIPSTART");
  Serial.println("");
  String ConexaoUDP="AT+CIPSTART=\"" +
    TIPOCONECAOUDP      + "\",\""  +
    IPSERVIDORUDP       + "\"," +
    PORTASERVIDORUDP    + "," +
    PORTACONTROLADORAUDP + ",2";
}
if (CMDSerial == 7) {
  Serial.println("Comando AT+CIPSEND");
  Serial.println("");
  String Envio="AT+CIPSEND=13";
  executaCMD(Envio);
}
if (CMDSerial == 8) {
  String Envio="AT+CIPSEND=13";
  executaCMD(Envio);
```

```
      delay(250);
      Envio="RC10101010101";
      executaCMD(Envio);
      delay(200);
      Serial.println("Enviado Comando RC10101010101 ");
      Serial.print("Para IP: " + IPSERVIDORUDP);
      Serial.println(" na Porta: " + PORTASERVIDORUDP);
      Serial.println("");
    }
    if (CMDSerial == 9) {
      Serial.println("Comando +IPD");
      Serial.println("");
      delay(100);
      String Envio="+IPD";
      executaCMD(Envio);
    }
    if (CMDSerial == 10) {
      CMDSerial=0;
      Serial.println("Comando AT+RST");
      Serial.println("");
      executaCMD("AT+RST");
    }
  }
```

Passo 03

Montagem dos Itens na Protobord.

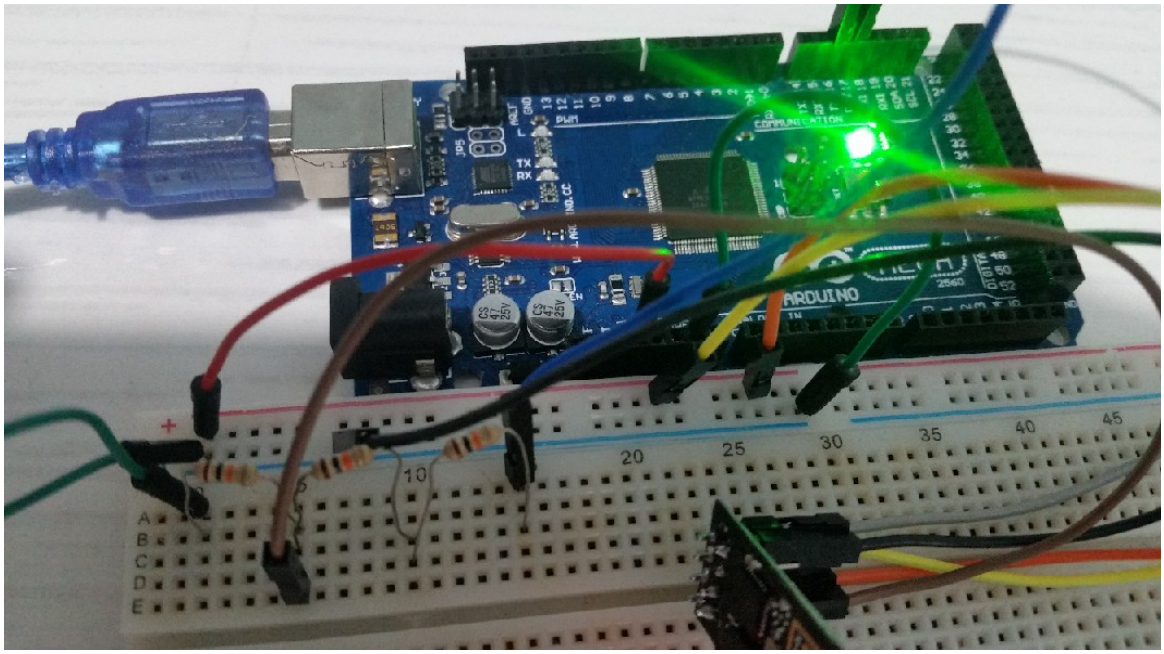
ESP8266	Controladora Arduino Mega
TX	Pino 17 (RX)
CH_PD	Pino 3,3V (Protobord)
RST	
3,3V	Pino 3,3V (Protobord)
GND	Pino GND (Protobord)
GPIO2	
GPIO0	
RX	Divisor de Tensão Entre (2,5V e 3.3V) → Pino 16 (TX)



Nota:

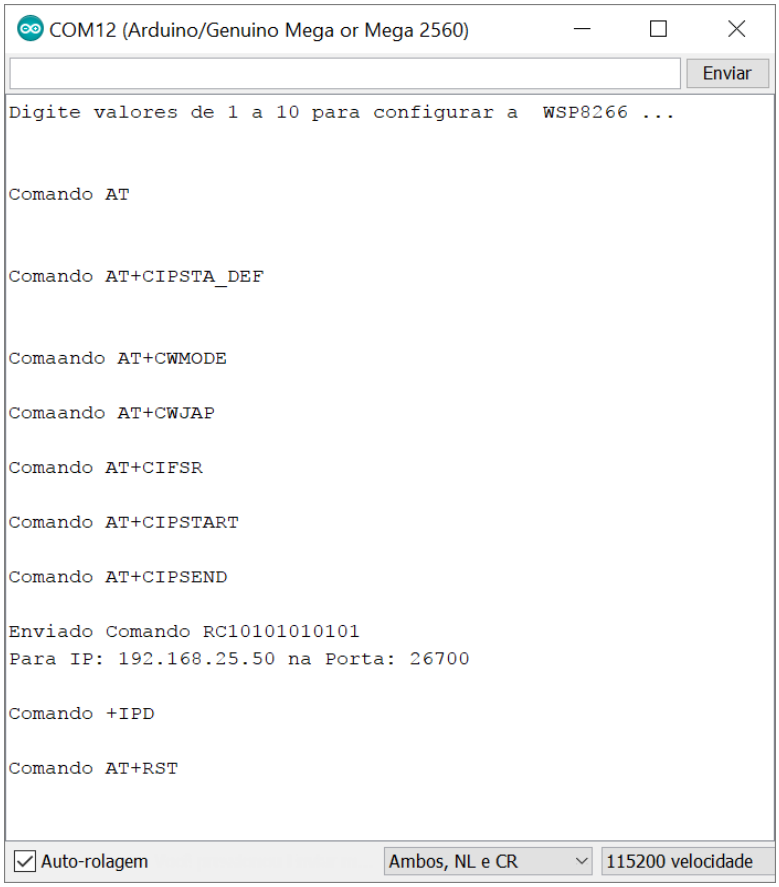
Tente variar entre 2,3V e 3,3V na sequência de resistores em sequência, para que consiga configurar seu módulo. O meu funcionou quando um pouco abaixo de 3,0V.

Confira na foto em funcionamento.



Passo 04

Utilizando o Monitor Serial da IDE do Arduino, digite pausadmanete de 1 a 10, para que seu módulo ESP8266 assuma os valores de IP e Porta, tanto para o módulo quanto para acesso ao serviço DataSnap. Mas antes, volte ao código fonte do sistema a ser ebarcado e altere os Ips para realidade de sua rede WIF, também, as credenciais de sua rede WIFI para possibilitar a conexão. Carregue o programa na Controladora e estabelaçõ oteste.



Repare a resposta no Monitor Serial após a entrada de valores de 1 a 10, pausadamente, um a um. Se tudo correu bem, você pode fazer um ping no IP que definiu para seu Módulo ESP8266. Também pode executar o aplicativo DataSnap, percebendo depois que digitou o comando 8 no Monitor Serial da IDE do Arduino, a entrada do valor RC10101010101, sinalizando que a controladora, via módulo ESP8266, utilizando a rede WIFI, conseguiu enviar para o sistema DataSnap, escutando via UDP, uma mensagem.

Contatos:

Laercio Guerço Rodrigues

laercio@itsoluction.com.br

21 99235 8440

Skype: laercio_guerco