```
program robuster_Ansatz;

{$ifdef MSWINDOWS}{$apptype CONSOLE}{$endif}
{$ifdef FPC}{$mode OBJFPC}{$H+}{$endif}

uses
 {$IFDEF UNIX}
  cthreads,
 {$ENDIF}
  SysUtils,
  IPConnection,
  Device,
  BrickletTemperatureV2,
  BrickletLCD128x64;

type
  TExampleRugged = class
  private
    ipcon: TIPConnection;
    TEMP_V2: TBrickletTemperatureV2;
    LCD_128x64: TBrickletLCD128x64;
  public
    procedure TemperatureCB(Sender: TBrickletTemperatureV2;
      const temperature: smallint);
    procedure EnumerateCB(Sender: TIPConnection; const uid: string;
      const connectedUid: string; const position: char;
      const hardwareVersion: TVersionNumber;
      const firmwareVersion: TVersionNumber;
      const deviceIdentifier: word; const enumerationType: byte);
    procedure Execute;
  end;

const
  HOST = 'localhost';
  PORT = 4223;
  TempV2UID = 'KEd';
  LCD128x64UID = '21q9';

var
  e: TExampleRugged;

  {Callback procedure for temperature and output to LCD_128_64 }
  procedure TExampleRugged.TemperatureCB(Sender: TBrickletTemperatureV2;
  const temperature: smallint);

  begin
    if Assigned(LCD_128x64) then
    begin
      LCD_128x64.ClearDisplay;
      LCD_128x64.WriteLine(0, 0,
        Format('Temperatur: %f', [temperature / 100.0]) + char(248) + 'C');
    end;
  end;

  procedure TExampleRugged.EnumerateCB(Sender: TIPConnection;
  const uid: string; const connectedUid: string; const position: char;
  const hardwareVersion: TVersionNumber;
  const firmwareVersion: TVersionNumber; const deviceIdentifier: word;
  const enumerationType: byte);
  begin
    if (enumerationType = IPCON_ENUMERATION_TYPE_CONNECTED)
    or (enumerationType = IPCON_ENUMERATION_TYPE_AVAILABLE)
    or (enumerationType <> IPCON_ENUMERATION_TYPE_DISCONNECTED)  then
    begin
      if deviceIdentifier = Bricklet_temperature_v2_device_identifier then
      begin
        if not Assigned(Temp_V2) then
        begin

          { Create device Object }
          TEMP_V2 := TBrickletTemperatureV2.Create(TempV2UID, ipcon);

          { Register temperature callback to proceduer TemperatureCB }
```

```
          TEMP_V2.OnTemperature := {$ifdef FPC}@{$endif}TemperatureCB;
        end;

        { set period for temperature callback to 1s without teshold }
        TEMP_V2.SetTemperatureCallbackConfiguration(1000, False, 'x', 0, 0);
      end;

      if deviceIdentifier = BRICKLET_LCD_128X64_DEVICE_IDENTIFIER then
        if not Assigned(LCD_128x64) then

          { Create device Object }
          LCD_128x64 := TBrickletLCD128x64.Create(LCD128x64UID, ipcon);
    end;
  end;

  procedure TExampleRugged.Execute;
  begin
    { Create connection }
    ipcon := TIPConnection.Create;

    { Register enumerate callback to "EnumerateCB" }
    ipcon.OnEnumerate := {$ifdef FPC}@{$endif}EnumerateCB;

    { Connect to brickd }
    ipcon.Connect(HOST, PORT);

    { Don't use device before ipcon is connected }

    { Trigger enumerate }
    ipcon.Enumerate;

    WriteLn('Press Return/Enter key to exit');
    ReadLn;

    LCD_128x64.Reset;

    { Calls LCD_128x64.Free internally }
    FreeAndNil(LCD_128x64);

    { Calls TEMP_V2.Free internally }
    FreeAndNil(TEMP_V2);

    { Calls ipcon.Free internally }
    FreeAndNil(ipcon);
  end;

begin
  e := TExampleRugged.Create;
  e.Execute;
  e.Free;
end.
```