MuntsOS Embedded Linux

Application Note #12: C# LED Flash Example

Revision 11
14 February 2023

by Philip Munts
President, Munts Technologies
http://tech.munts.com

Introduction

This application note describes how to create, build, and run a C# program to flash an LED on a target computer running ***MuntsOS Embedded Linux***, using the ***.Net*** SDK (Software Development Kit).

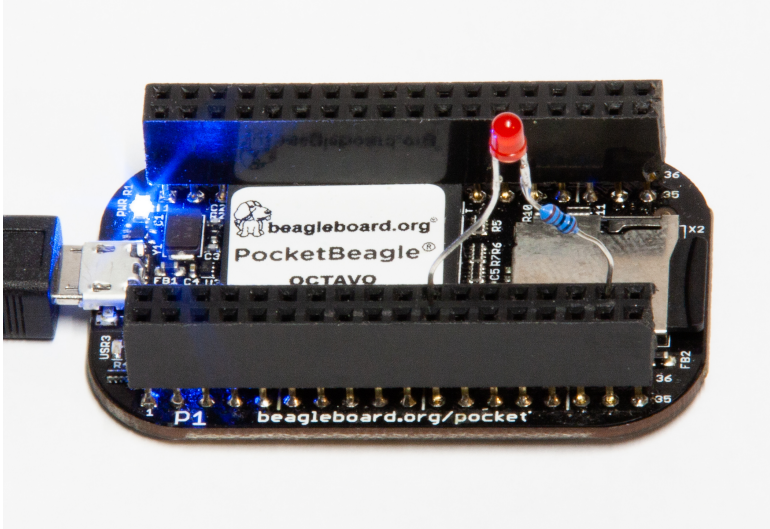
Prerequisites

The ***.Net*** SDK must be installed on a Linux, MacOS, or Windows host system.

MuntsOS Embedded Linux must be installed on the target computer and USB Network Gadget mode configured (AppNote #11).

The ***.Net*** runtime extension **`dotnet-muntsos-BeagleBone.deb`** must be installed on the target computer.

Test Platform Hardware



Unfortunately, the ***.Net*** runtime is not available for the ARMv6 instruction set of the Raspberry Pi 1 and Zero boards. It requires the 32-bit ARMv7 instruction set (any BeagleBone or Raspberry Pi 2 or 3) or the 64-bit ARMv8 instruction set (Raspberry Pi 3 or 4).

The test platform for the purposes of this application node consists of a PocketBeagle with female headers mounted on top. The PocketBeagle is a small, reduced cost member of the BeagleBone family that targets the same market niche as the lower cost Raspberry Pi Zero. It is likely the smallest platform that can run ***.Net*** applications.

An LED with a 1 kilohm current limiting resistor is plugged in to the PocketBeagle header, from P1.34 (**`GPIO26`**) to P1.22 (**`GND`**).

Test Program Source Code

Available for download at: http://git.munts.com/muntsos/doc/.blinky/blinky.cs

```
using static System.Console;

WriteLine("\nMuntsOS C# LED Test\n");

// Configure a GPIO output to drive an LED

IO.Objects.libsimpleio.Device.Designator desg_LED =
  new IO.Objects.libsimpleio.Device.Designator(0, 26);

IO.Interfaces.GPIO.Pin LED =
  new IO.Objects.libsimpleio.GPIO.Pin(desg_LED,
    IO.Interfaces.GPIO.Direction.Output);

// Flash the LED forever (until killed)

WriteLine("Press CONTROL-C to exit.\n");

for (;;)
{
  LED.state = !LED.state;
  System.Threading.Thread.Sleep(500);
}
```

Exercise

This example exercise demonstrates how to create a C# program project, compile it with the ***.Net*** SDK, and run it on the test platform hardware.

Step 1: Install ***.Net*** project templates for the ***Linux Simple I/O Library***, from https://www.nuget.org :

```
dotnet new install libsimpleio-templates
```

Step 2: Create a ***.Net*** console application project:

```
dotnet new csharp_console_libsimpleio -o blinky
cd blinky
```

Step 3: Replace **`Program.cs`** in the project with **`blinky.cs`**:

```
wget -O Program.cs http://git.munts.com/muntsos/doc/.blinky/blinky.cs
```

Step 4: Compile the application:

```
dotnet publish
```

Step 5: Copy the program files to the target platform:

```
scp bin/Debug/net6.0/publish/* root@usbgadget.munts.net:.
```

Step 6: Run the test program on the test platform:

```
ssh root@usbgadget.munts.net
dotnet blinky.dll
```

The LED should begin flashing once a second, until you press **`CONTROL-C`**.