MuntsOS Embedded Linux

Application Note #14: Modula-2 LED Flash Example

Revision 3
19 February 2025

by Philip Munts
***dba* Munts Technologies**
http://tech.munts.com

Introduction

This application note describes how to create, build, and run a GNU Modula-2 program to flash an LED on a target computer running ***MuntsOS Embedded Linux***.

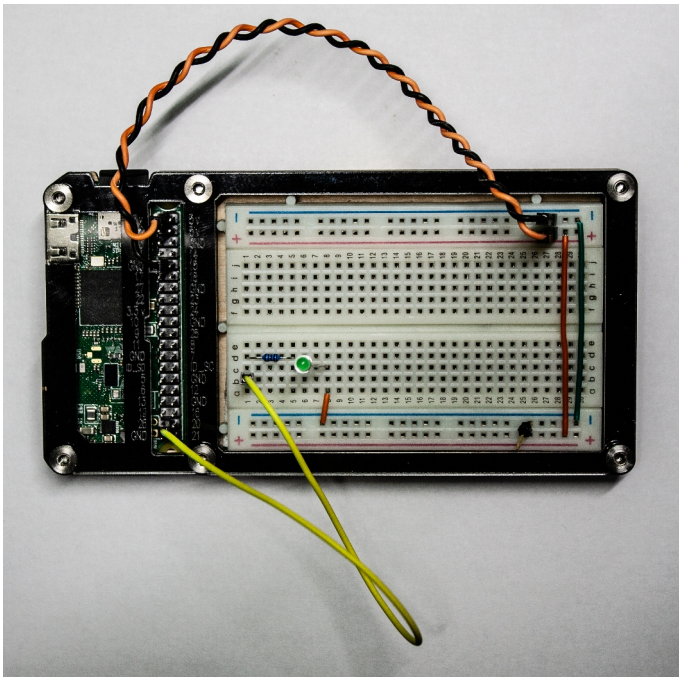
Prerequisites

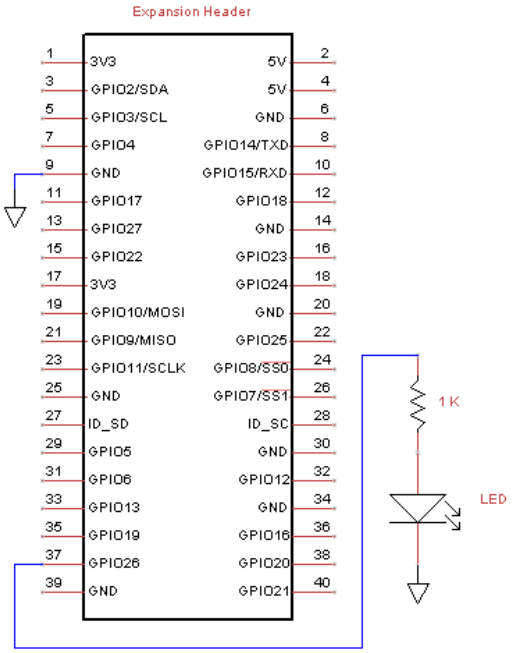
The ***MuntsOS Embedded Linux*** software development environment must be installed on a Linux development computer (AppNote #1 or AppNote #2).

MuntsOS Embedded Linux must be installed on the target computer (AppNote #3).

The GNU Modula-2 shared library extension **`libgm2-muntsos-aarch64.deb`** must be installed on the target computer, by running the **`sysconfig`** command on the target computer.

Test Platform Hardware





The test platform for the purposes of this application note consists of a Raspberry Pi Zero 2 Wireless mounted in a Zebra Zero Plus Breadboard case. The orange and black jumper wires connect **`+3.3V`** and **`GND`** on the Raspberry Pi expansion header to the breadboard power rails. The yellow jumber connects **`GPIO26`** to a 1K ohm current limiting resistor and an LED.

Test Program Source Code

Available for download at: https://repo.munts.com/muntsos/doc/.blinky/blinky.mod

```
MODULE blinky;

IMPORT GPIO_libsimpleio, RaspberryPi;

FROM STextIO       IMPORT WriteString, WriteLn;
FROM FIO           IMPORT FlushOutErr;
FROM ErrorHandling IMPORT CheckError;
FROM liblinux      IMPORT Sleep;

VAR
  LED      : GPIO_libsimpleio.Pin;
  error    : CARDINAL;
  state    : BOOLEAN;

BEGIN
  WriteLn;
  WriteString("Raspberry Pi LED Test");
  WriteLn;
  WriteLn;
  FlushOutErr;

  (* Configure LED GPIO output *)

  GPIO_libsimpleio.Open(RaspberryPi.GPIO26, GPIO_libsimpleio.Output, FALSE,
    GPIO_libsimpleio.PushPull, GPIO_libsimpleio.None,
    GPIO_libsimpleio.ActiveHigh, LED, error);
  CheckError(error, "GPIO_libsimpleio.Open() failed");

  WriteString("Press CONTROL-C to exit");
  WriteLn;
  WriteLn;
  FlushOutErr;

  LOOP
    GPIO_libsimpleio.Read(LED, state, error);
    CheckError(error, "GPIO_libsimpleio.Read() failed");

    GPIO_libsimpleio.Write(LED, NOT state, error);
    CheckError(error, "GPIO_libsimpleio.Write() failed");

    Sleep(500000, error);
    CheckError(error, "LINUX_usleep() failed");
  END;
END blinky.
```

Exercise

This example exercise demonstrates how to create a GNU Modula-2 program project, build it, and run it on the test platform hardware.

Step 1: Prepare the **blinky** project:

```
mkdir $HOME/blinky
cd $HOME/blinky
wget https://repo.munts.com/muntsos/doc/.blinky/Makefile.gm2
mv Makefile.gm2 Makefile
wget https://repo.munts.com/muntsos/doc/.blinky/blinky.mod
```

Step 2: Build the **blinky** project:

```
make BOARDNAME=RaspberryPiZero2W
```

Step 3: Copy **blinky** to the test platform:

```
scp blinky root@snoopy:.
```

Step 4: Run the test program on the test platform:

```
ssh root@snoopy
./blinky
```

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