

# **MuntsOS Embedded Linux**

## ***Application Note #5: C++ LED Flash Example***

**Revision 3  
26 June 2019**

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

## Introduction

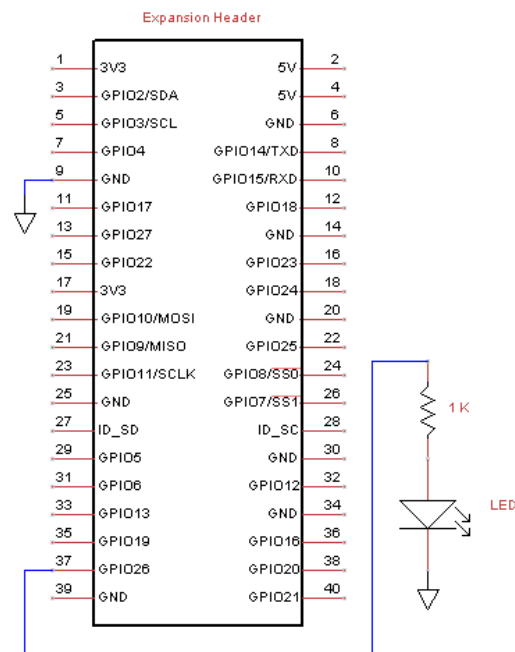
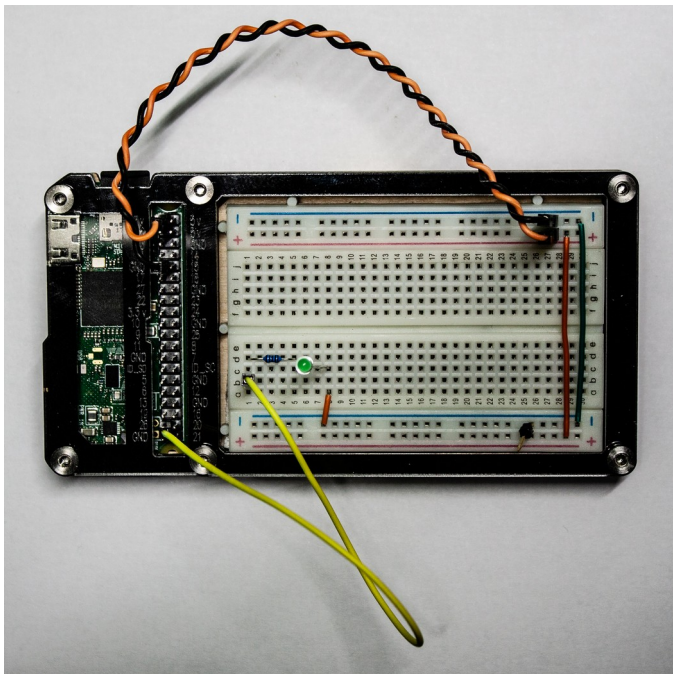
This application note describes how to create, build, and run a C++ program to flash an LED on a target computer running **MuntsOS Embedded Linux**.

## Prerequisites

The **MuntsOS Embedded Linux** software development environment must be installed on a 64-bit x86-64 Linux system ([AppNote #1](#) or [AppNote #2](#)).

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

## Test Platform Hardware



The test platform for the purposes of this application note consists of a [Raspberry Pi Zero 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 jumper connects GPIO26 to a 1 K ohm current limiting resistor and an LED.

## Test Program Source Code

Available for download at: <http://git.munts.com/muntsos/doc/blink/blink.cpp>

```
#include <cstdio>
#include <unistd.h>

#include <raspberrypi.h>

int main(void)
{
    puts("\nMuntsOS C++ LED Test\n");

    // Configure a GPIO output to drive an LED

    Interfaces::GPIO::Pin LED =
        new libsimpleio::GPIO::Pin_Class(RaspberryPi::GPIO26,
            Interfaces::GPIO::OUTPUT, false);

    // Flash the LED forever (until killed)

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

    for (;;)
    {
        *LED = ! *LED;
        usleep(500000); // microseconds = 0.5 seconds
    }
}
```

## **Exercise**

This example exercise demonstrates how to create a C++ program project (outside of the **MuntsOS** code tree checkout), compile it, and run it on the test platform hardware.

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

```
mkdir $HOME/blinky
cd $HOME/blinky
cp $HOME/muntsos/doc/.blinky/Makefile.c++ Makefile
cp $HOME/muntsos/doc/.blinky/blinky.cpp .
```

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

```
make BOARDNAME=RaspberryPi1
```

*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**.