

Using the Texas Instrument MSP 430 Launchpad with Linux

Gareth Digby
for
The Columbia Area Linux User Group (CALUG)

May 13th, 2015



Copyright © Gareth Digby 2015

Introduction

- ❖ Background
- ❖ Texas Instruments MSP 430 Launchpad
- ❖ Setting up Linux for cross platform development
- ❖ Development cycle
- ❖ Demonstration(s)



Copyright © Gareth Digby 2015

Using the Texas Instrument MSP 430 Launchpad
with Linux

May 13th, 2015
Page 2

Background

- ❖ The MSP 430 and a Linux cross platform development setup were the subjects of an assignment I wrote for the forensics course I teach.
 - ❖ “The Ministry of Defence decided to upgrade the flashing red light on the UK’s Avro Vulcan strategic bomber fleet. A contract was awarded to Bodge It & Scarper of Maryland, USA, for the development of an embedded device, using the Texas Instruments MSP 430, to control the flashing light. However a number of failures in flight have led to the grounding of the aircraft fleet while the quality of work is investigated.”
 - ❖ “Evidence has been recovered from the embedded controllers on five aircraft.”
 - ❖ “During the week of January 27th – 31st, Mr. Neddy Seagoon was employed as a software developer by Bodge It & Scarper. He is believed to have been involved in developing the software for the MSP 430 embedded device. Evidence has been recovered from Mr. Seagoon’s computer.”



Copyright © Gareth Digby 2015

Using the Texas Instrument MSP 430 Launchpad
with Linux

May 13th, 2015
Page 3

Avro Vulcan



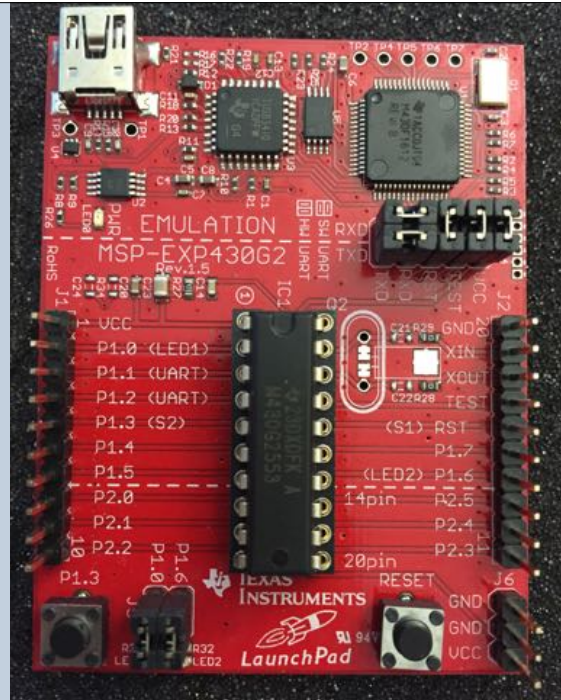
Copyright © Gareth Digby 2015

Using the Texas Instrument MSP 430 Launchpad
with Linux

May 13th, 2015
Page 4

Texas Instruments MSP 430 Launchpad

- ✦ MSP430 LaunchPad includes everything you need to get started with MSP430 development!
 - ✦ MSP-EXP430G2 LaunchPad
 - ✦ MSP430G2553 microcontroller (with demo code)
 - ✦ MSP430G2452 microcontroller
 - ✦ Mini-USB cable
 - ✦ Quick Start Guide
 - ✦ 32kHz external crystal

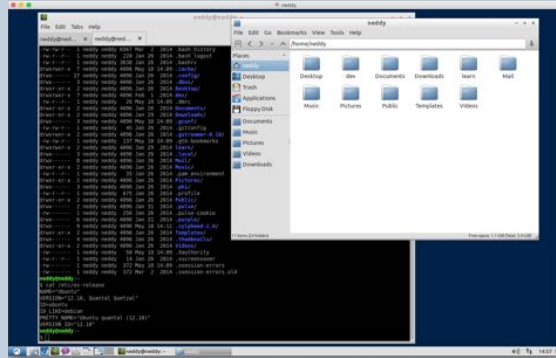


TI MSP 430G2553

- ✦ Microcontroller Features
 - ✦ 16MHz
 - ✦ 16KB Flash
 - ✦ 512B RAM
 - ✦ 8ch 10-bit ADC
 - ✦ Comparator
 - ✦ 2 x 16-bit Timers
 - ✦ Up to 1 x I2C, 2 x SPI, 1 x UART

Linux Used For Cross Platform Development

- ❖ Ubuntu 12.10
- ❖ Ubuntu Quantal (12.10)



Copyright © Gareth Digby 2015

Using the Texas Instrument MSP 430 Launchpad
with Linux

May 13th, 2015
Page 7

Linux Development Environment

- ❖ nano - Nano's ANOther text editor
- ❖ make - GNU make utility to maintain groups of programs
- ❖ mspdebug - debugging and programming tool for MSP430 MCUs
- ❖ Along with:
 - ❖ msp430-libc
 - ❖ mspdebug
 - ❖ msp430mcu
 - ❖ binutils-msp430
 - ❖ gcc-msp430
 - ❖ gdb-msp430
- ❖ minicom - friendly serial communication program



Copyright © Gareth Digby 2015

Using the Texas Instrument MSP 430 Launchpad
with Linux

May 13th, 2015
Page 8

Setting Up Linux

```
$ sudo apt-get install build-essential
$ sudo apt-get install linux-headers-$(uname -r)

$ sudo apt-get install git
$ git config --global user.email user@example.com

$ sudo apt-get install msp430-libc mspdebug
msp430mcu binutils-msp430 gcc-msp430 gdb-msp430

$ PATH=$PATH:/usr/msp430/lib/ldscripts/msp430g2553/
```



Setting Up Linux

- * Makefile
 - * include in folder where code will be compiled
 - * gcc cross compiles for the MSP 430
 - * object file is named main.elf

```
CC=msp430-gcc
CFLAGS=-Os -Wall -g -mmcu=msp430g2553
# CFLAGS=-Os -Wc++ -g -mmcu=MCU
OBJS=main.o
all: $(OBJS)
    $(CC) $(CFLAGS) -o main.elf $(OBJS)
%.o: %.c
    $(CC) $(CFLAGS) -c $<
clean:
    rm -fr main.elf $(OBJS)
```



Development Cycle

- ❖ Edit C code
 - nano
- ❖ Cross compile/build code on Linux for MSP 430
 - make
- ❖ Identify USB bus MSP 430 is connected to
 - lsusb
- ❖ Load code into MSP 430
 - mspdebug -U 002:006 rf2500 "prog main.elf"



Development Cycle - edit C code

- ❖ Edit C code
 - nano
- ❖ Cross compile/build code on Linux for MSP 430
 - make
- ❖ Identify USB bus MSP 430 is connected to
 - lsusb
- ❖ Load code into MSP 430
 - mspdebug -U 002:

```
nanoddy@ceddy: ~/dev/blink
File Edit Tabs Help
nanoddy@ceddy: ~$ nano ~/dev/blink.c
~/dev/blink.c
// Built with IWR Embedded Workbench version: 1.004
//-----
#include "msp430.h"

int main(void)
{
    WDTCTL = WDTPW + WDTHOLD; // Stop watchdog timer
    P1DIR |= 0x01; // set P1.0 to output direction
    P1OUT |= 0x01; // set P1.0 to output direction

    // Turn on the first LED, the other stays off
    P1OUT = 0x01; // set P1.0 on (red led)

    for (;;)
    {
        volatile unsigned int i; // volatile to prevent optimization
        //Main toggle
        //P1OUT ^= 0x01; // toggle P1.0 using exclusive-OR

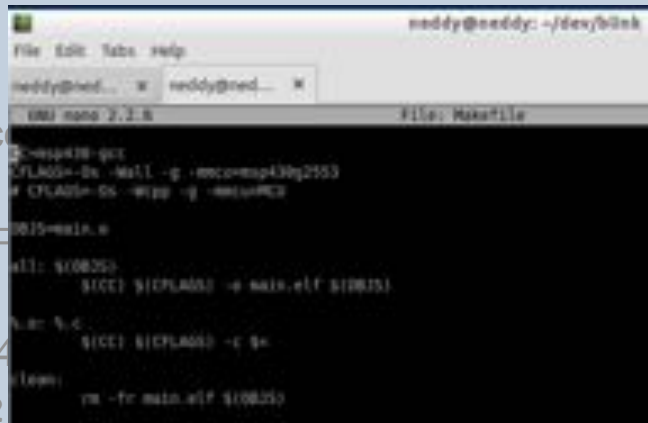
        // This option uses bit shifts. It looks a little more
        // complicated but is optimized by the compiler so it's
        // fast and efficient.
        P1OUT ^= (1 | (1 << 0)); // Common approach

        i = 10000; // 10 Delay
        do {
            while (i > 0);
        } while (i > 0);
    }
}
```



Development Cycle - cross compile code

- ❖ Edit C code
nano
- ❖ Cross compile/build code
make
- ❖ Identify USB bus MSP430
lsusb
- ❖ Load code into MSP430
mspdebug -U 002

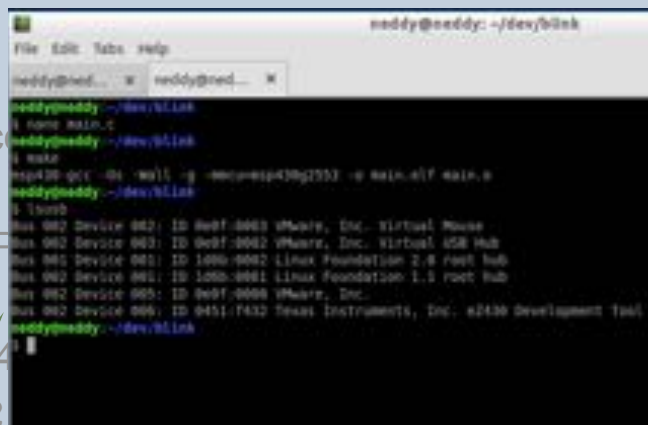


```
neddy@neddy: ~/dev/blink
File Edit Tabs Help
neddy@ned... x neddy@ned... x
GNU nano 2.2.6 File: Makefile
$-gcc430-gcc
CFLAGS=-Dx_Mall -g -mcpu=msp430g2553
CFLAGS=-Dx -fPIC -g -mcpu=M3
0025=main.c
all: $(OBJS)
$(CC) $(CFLAGS) -o main.o $(OBJS)
%.o: %.c
$(CC) $(CFLAGS) -c %
clean:
rm -fr main.o $(OBJS)
```



Development Cycle - identify USB device

- ❖ Edit C code
nano
 - ❖ Cross compile/build code
make
 - ❖ Identify USB bus MSP430
lsusb
 - ❖ Load code into MSP430
mspdebug -U 002
- USB bus 002 device 006 02



```
neddy@neddy: ~/dev/blink
File Edit Tabs Help
neddy@ned... x neddy@ned... x
neddy@neddy:~/dev/blink
$ nano main.c
neddy@neddy:~/dev/blink
$ make
msp430-gcc -Dx_Mall -g -mcpu=msp430g2553 -o main.o main.c
neddy@neddy:~/dev/blink
$ lsusb
Bus 002 Device 002: ID 040f:3003 VMware, Inc. Virtual Mouse
Bus 002 Device 002: ID 040f:3002 VMware, Inc. Virtual USB Hub
Bus 002 Device 001: ID 1098:0002 Linux Foundation 2.8 root hub
Bus 002 Device 001: ID 1098:0001 Linux Foundation 1.1 root hub
Bus 002 Device 005: ID 040f:9004 VMware, Inc.
Bus 002 Device 006: ID 0451:7432 Texas Instruments, Inc. MSP430 Development Tool
neddy@neddy:~/dev/blink
```



Development Cycle

- ❖ Edit C code

nano

- ❖ Cross compile/build code

make

- ❖ Load code into MSP430

lsusb

- ❖ Erase & reprogram device with binary file

mspdebug -U 002:006 rf2500 "prog main.elf"

driver to connect via USB

erase & reprogram device with binary file main.elf

```
Bus 002 Device 006: ID 0451:1432 Texas Instruments, Inc. e2430 Development Tool
maddy@maddy:~/dev/blink
maddy@maddy:~/dev/blink
$ mspdebug -U 002:006 rf2500 "prog main.elf"
mspdebug version 0.29 - debugging tool for MSP430 MCUs
Copyright (C) 2009-2012 Daniel Beer <d@beer@gmail.com>
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Trying to open interface 1 on 006.
Initializing FET...
FET protocol version is 30394238
Configured for Spy-BI-Wire
Get Vcc: 3000 mV
Device ID: 8a2553
Device: MSP430G2553
Code memory starts at 0xc000
Number of breakpoints: 2
Erasing...
Programming...
Writing 128 bytes to 0000 [section: .text]...
Writing 32 bytes to f000 [section: .vectors]...
Done, 160 bytes written
maddy@maddy:~/dev/blink
$
```



Debugging

- ❖ Use mspdebug to debug applications

mspdebug -U 002:006 rf2500

```
maddy@maddy:~/dev/blink
File Edit Tabs Help
maddy@maddy:~/dev/blink
maddy@maddy:~/dev/blink
maddy@maddy:~/dev/blink
$ mspdebug -U 002:006 rf2500
mspdebug version 0.29 - debugging tool for MSP430 MCUs
Copyright (C) 2009-2012 Daniel Beer <d@beer@gmail.com>
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Trying to open interface 1 on 006.
rf2500: warning: can't detach named driver: No data available
Initializing FET...
FET protocol version is 30394238
Configured for Spy-BI-Wire
Get Vcc: 3000 mV
FET: FET returned error code 4 (Could not find device (or device not supported)
FET: command C IDENTI Failed
FET: identify failed
Trying again...
Initializing FET...
FET protocol version is 30394238
Configured for Spy-BI-Wire
Sending reset...
Get Vcc: 3000 mV
Device ID: 8a2553
Device: MSP430G2553
Code memory starts at 0xc000
Number of breakpoints: 2

Available commands:
  w      delbreak  gdb      load     oph      reset    size
 alias  dis       help    locks   prog    run      step
 break  erase     hexout  md      read    set      sym
 tgraph  xkill    search  sw      regs   setbreak

Available options:
 color      gdb  time  lradix
 fet_block_size  gdbc_xfer_size  quiet

Type "help <topic>" for more information.
Press Ctrl-Q to quit.
mspdebug: |
```



Demonstration

- * Examples
 - * blink
 - * flash the LEDs
 - * uart_rx
 - * toggle LEDs when characters received by the UART
 - * uart_rx_tx
 - * send back text when characters received by the UART
 - * uart_hardware
 - * switch LEDs on/off depending character received via UART



Demonstration - blink

- * Edit C code

```
nano blink.c
```
- * Cross compile/build code on Linux for MSP 430

```
make
```
- * Identify USB bus MSP 430 is connected to

```
lsusb
```
- * Load code into MSP 430

```
mspdebug -U 002:006 rf2500 "prog main.elf"
```



Demonstration - uart_rx

```
nano blink.c
make
lsusb
mspdebug -U 002:006 rf2500 "prog main.elf"
* Run minicom to transmit to MSP 430 UART via USB
sudo minicom -b 9600 -D /dev/ttyACM0
```

- * press A to toggle red LED
- * press B to toggle green LED
- * ctrl-a q to exit minicom



Demonstration - uart_rx_tx

```
nano blink.c
make
lsusb
mspdebug -U 002:006 rf2500 "prog main.elf"
sudo minicom -b 9600 -D /dev/ttyACM0
```

- * press A to receive "Hello World"
 - * red led lights briefly on transmit
 - * green led lights briefly on receive
- * ctrl-a q to exit minicom



Demonstration - uart hardware

```
nano blink.c
make
lsusb
mspdebug -U 002:006 rf2500 "prog main.elf"
sudo minicom -b 9600 -D /dev/ttyACM0
```

- * press R to turn on red LED
- * press r to turn off red LED
- * press G to turn on green LED
- * press g to turn off green LED
- * ctrl-a q to exit minicom



Copyright © Gareth Digby 2015

Using the Texas Instrument MSP 430 Launchpad with Linux May 13th, 2015 Page 21

Bibliography

- * Vulcan Image
 - * http://commons.wikimedia.org/wiki/File:Avro_Vulcan_Bomber_RAF.JPG
 - * Sgt. David S. Nolan, US Air Force
- * MSP 430 Information
 - * <http://www.ti.com/launchpad>
 - * <http://www.ti.com/ww/en/launchpad/launchpads-msp430-msp-exp430g2.html#tabs>
- * Blinking LED demo
 - * http://hcoop.net/~t0rch/LaunchPad/msp430x2xx_fet_1.c
- * UART RX demo
 - * <http://forum.43oh.com/topic/1764-msp430g2553launchpad-uart-rx-example/>
- * UART RX TX demo
 - * <http://www.embeddedrelated.com/showarticle/420.php>
- * UART RX TX hardware tutorial
 - * <http://benntonsen.wordpress.com/ti-msp430-launchpad/msp430g2553-hardware-uart/>
- * To compile on different versions of MSP 430
 - * <http://justintech.org/2011/07/msp430-different-interrupts-for-different-compilers/>



Copyright © Gareth Digby 2015

Using the Texas Instrument MSP 430 Launchpad with Linux May 13th, 2015 Page 22

If USB Does Not Work Install Kernel Patch

- ✦ Download `kernel_patch.zip` from:
http://e2e.ti.com/cfs-file.ashx/___key/communityserver-discussions-components-files/156/7028.msp430_5F00_patch.zip
- ✦ Install USB kernel driver, unzip `kernel_patch.zip` and then:

```
cd kernel_patch/  
ls  
cd linux-3.0/  
cd cdcacm-0.1/  
make clean  
make  
sudo modprobe -r cdc_adm  
sudo insmod ./cdcacm.ko
```



Copyright © Gareth Digby 2015

Using the Texas Instrument MSP 430 Launchpad
with Linux

May 13th, 2015
Page 23

License

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.



Copyright © Gareth Digby 2015

Using the Texas Instrument MSP 430 Launchpad
with Linux

May 13th, 2015
Page 24