Small Computers Community Workshop

January 25, 2017

Contents

- Quick intro to ‘small computers’
- Next Thing Co CHIP single board computer
- CHIP specs and some things I’ve learned using it
- How we are using CHIPs and Raspberry Pis with Nexus
- Compare CHIP with Arduino Uno
- Show how to control an LED with CHIP vs. Arduino
- Focus will be on ‘headless’ usage

Small Computers

- There are lots (100s?) of small computers available, many options
- ‘System on a Chip’ with full operating system or microcontroller
- Communications hardware (Wifi, Bluetooth, NFC, ethernet)
- I/O pins and electrical characteristics (how many, current source and sink, 3.3V or 5V, can you drive an LED, or a motor)
- Analog to digital converters (in), digital to analog converters (out)
- On-board sensors and outputs such as LEDs or buzzers
- Availability of add-on boards: HATs, shields, wings, DIPs
- Power consumption (measurement of Micro:bit power consumption)

Some Small Computers (very non-exhaustive)

- Raspberry Pi (2012)
- CHIP (2015/2016)
- Arduino (2005)
- Small Arduino compatibles: Teensy and Adafruit Feather
- Wearable Arduino compatibles: LilyPad and Adafruit FLORA
- Micro:bit
- Beaglebone Black
- Intel Edison and Intel Joule

System on a Chip Operating Systems

- Full Linux distribution
  - For example: Debian on CHIP or Raspbian on Raspberry Pi
- Cross-compiling embedded Linux distribution
  - For example: Buildroot (CHIP supported) or Yocto
- Android Things
- Windows IoT
- Will focus on Linux today

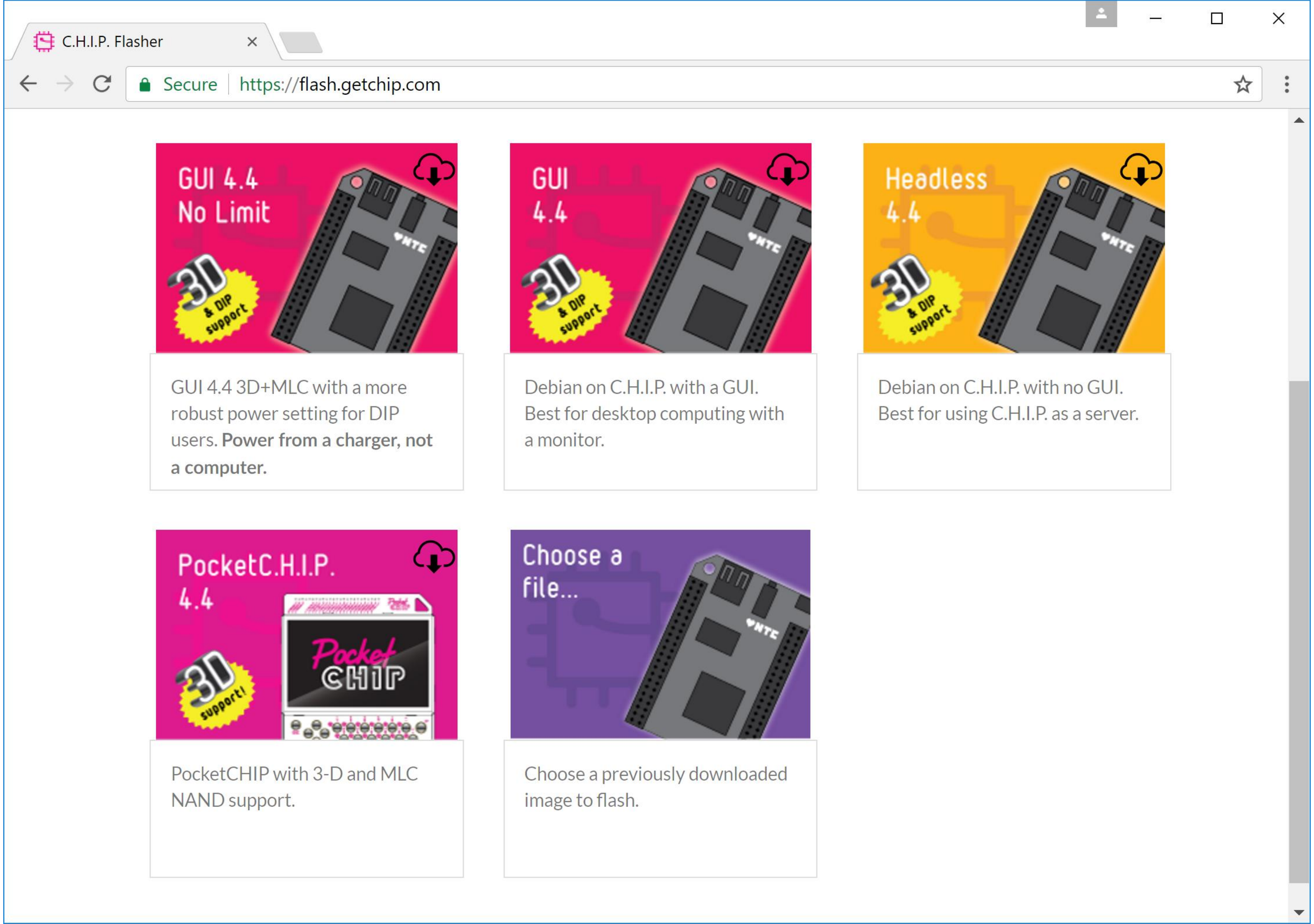
CHIP Specs

- 1 GHz ARM v7 Allwinner R8 CPU
- 512MB DRAM
- 4GB NAND storage (UBIFS)
- Realtek RTL8723BS
  - 802.11b/g/n (`wlan0` and `wlan1`)
  - Bluetooth 4.0 LE
- https://docs.getchip.com/chip.html#parts-and-pieces
- CHIP Hardware Datasheets

- USB Micro B socket power and USB On-The-Go
- USB A socket Host (500 mA)
- TRRS A/V connector
- 80 pin headers
- Battery connector
- Power button
- Open source hardware

Flashing CHIP

- Next Thing Co provides a Chrome App for flashing CHIP
- Use a jumper wire to connect the FEL pin to GND
- Connect the CHIP to a computer USB 2.0 socket (I needed to use an external hub on the Surface Pro 4)
- https://docs.getchip.com/chip.html#flash-chip-with-an-os
- https://flash.getchip.com/



GUI 4.4 3D+MLC with a more robust power setting for DIP users. **Power from a charger, not a computer.**

Debian on C.H.I.P. with a GUI. Best for desktop computing with a monitor.

Debian on C.H.I.P. with no GUI. Best for using C.H.I.P. as a server.

PocketCHIP with 3-D and MLC NAND support.

Choose a previously downloaded image to flash.

Powering CHIP

- USB Micro B
  - Connect to a computer USB host socket or to a USB power adapter
  - In the default image, current is limited
  - AXP209 Current Limit Settings
- Lithium-ion polymer (LiPo) Battery
  - Including battery charging
  - (more on using a LiPo battery later)
- CHG-IN pin
- http://www.chip-community.org/index.php/Power

Connecting to CHIP

- Serial USB to Micro B USB OTG
- UART: GND, TX, RX
  - For example, using the Adafruit FTDI Friend
  - See logging messages on boot and shutdown, including from U-Boot
  - Connection survives reboot and power cycling of CHIP
- SSH wireless or wired (wired requires USB ethernet adapter)
- https://docs.getchip.com/chip.html#headless-chip

CHIP Default User Accounts and SSH

- User: chip, password: chip
- User: root, password: chip
- On the headless image, the ssh service is enabled
- PermitRootLogin is “yes” in `/etc/ssh/sshd_config`
- Change password with: `passwd`
- Disable root login with: `sudo passwd -l root`
- Disable ssh service with: `sudo systemctl disable ssh`

CHIP Hostname

- Default hostname: chip
- To change the hostname, edit:
  - `/etc/hostname`
  - `/etc/hosts`
- The avahi-daemon service (Zeroconf multi-cast DNS) is running by default, so we can locate with HOSTNAME.local

Updating the Operating System (Debian)

```
$ sudo apt update
$ sudo apt upgrade
```

Connecting CHIP to a wireless network

- `wlan0` is controlled by NetworkManager; `wlan1` is not
- List available networks:

```
$ nmcli device wifi
```

- Connect:

```
$ sudo nmcli device wifi connect NETWORK ifname
wlan0
$ sudo nmcli connection up ifname wlan0
```

- Check connection status:

```
$ nmcli device status
```

CHIP and Lithium-ion Polymer (LiPo) Batteries

- CHIP has charging circuit for LiPo batteries
  - Single cell
  - 3.7V
  - JST-PH connector
- Check battery status with: `sudo battery.sh`
- https://docs.getchip.com/chip.html#power-up
- Adafruit article on Li-Ion and LiPoly Batteries
- SparkFun LiPo Battery Care tutorial

Nexus

- https://github.com/simonbates/nexus-demos/blob/master/docs/NexusOnCHIP.md
- https://github.com/simonbates/nexus-demos/blob/master/docs/BrlapiOnCHIP.md
- `wscat -c ws://nexuschip.local:9081/bindModel/nexus.brailleDisplay/displayText`
- `{ "path": "", "value": "hello" }`

Linux vs. Microcontrollers (generalizations)

Linux

- Use as ‘regular’ computer or cross-compiled
- Takes time for Linux to boot
- Risk of filesystem loss if not shutdown cleanly
- More complex and likely to need ongoing security patching
- Extensive driver support

Microcontrollers

- Cross-compile (image may be an interpreter such as MicroPython)
- Starts up very quickly
- Generally safe to just kill power
- Probably a better fit for real-time applications

CHIP and Arduino Uno

CHIP

- 1 GHz Allwinner R8 CPU
- 512MB DRAM, 4GB NAND storage
- Linux
- 1 x 6-bit Analog Digital Converter

Arduino Uno

- 20 MHz Atmel ATmega328 CPU
- 2 KB SRAM, 32 KB flash (program), 1 KB EEROM (non-volatile storage)
- No operating system (Arduino bootloader)
- 6 x 10-bit ADC
- I/O Pins can source up to 40 mA

Controlling an LED on Arduino

- https://www.arduino.cc/en/tutorial/blink

CHIP XIO GPIO Pins

- 8 pins provided by PCF8574 via I^{2}C
- Current when output is high (IOH): 30 to 300 µA
- Current when output is low (IOL): 10 to 25 mA
- http://www.ti.com/lit/ds/symlink/pcf8574.pdf

Controlling an LED on CHIP 1/3

- We can’t use the Arduino circuit on CHIP as the XIO pins cannot source enough current
- Instead, we can either use an XIO pin to control a current from another source (for example using a FET (Field-Effect Transistor))
- Or, ‘reverse’ the circuit and sink to an XIO pin rather than source

Controlling an LED on CHIP 2/3

- Circuit: VCC-3V3 --> LED --> resistor --> XIO-P0
- Resistor value = (supply voltage – LED forward voltage drop) / current
- Supply voltage = 3.3 V
- LED forward voltage drop = 1.7 V
- Current = 10 mA (0.01 A)
- Resistor = (3.3 - 1.7) / 0.01 = 160 Ohms (round up to 220 Ohms)
- SparkFun tutorial on selecting resistor value for LED circuit: https://www.sparkfun.com/tutorials/219

Controlling an LED on CHIP 3/3

```
$ sudo su -
$ cd /sys/class/gpio
$ echo 1013 > export
$ cat gpio1013/direction
$ echo out > gpio1013/direction
$ echo 0 > gpio1013/value
$ echo 1 > gpio1013/value
$ echo 1013 > unexport
```

CHIP GPIO Resources

- https://docs.getchip.com/chip.html#pin-headers
- GPIO device numbering varies with kernel version
- In the current kernel, 4.4.13-ntc-mlc, XIO-P0 is 1013 https://docs.getchip.com/chip.html#gpio
- C GPIO library: https://github.com/chip-community/libsoc
- Python GPIO library: https://github.com/xtacocorex/CHIP_IO
- https://bbs.nextthing.co/t/electrical-characteristics-of-chips-gpio/2590
- http://www.chip-community.org/index.php/GPIO_Info

Useful Accessories

- USB power adapter (1A to 2.5A)
- Powered USB Hub
  - http://elinux.org/RPi_Powered_USB_Hubs
- USB Ethernet adapter
- USB to serial (UART) cable or board
- Jumper wires
- Multimeter

Some CHIP Resources

- Official CHIP Documentation
- CHIP Community Wiki
- Talk by Dave Rauchwerk of Next Thing Co on the history and design of the CHIP (YouTube video 65 mins)
- CHIP Hardware Datasheets
- Learning the Basics of Buildroot (YouTube video 116 mins)