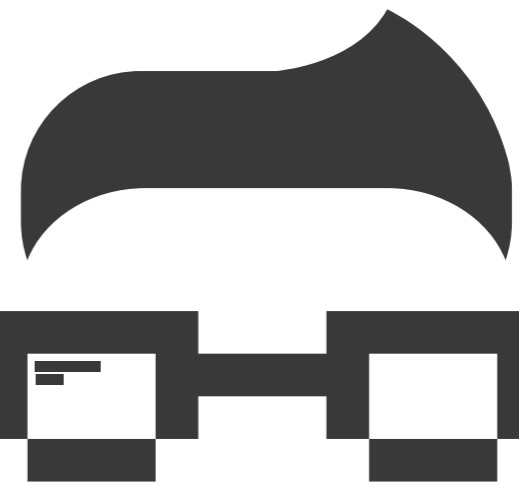


Accelerate design and prototyping of IoT solutions with PixiePro



treats
4GEEKS

by 
code



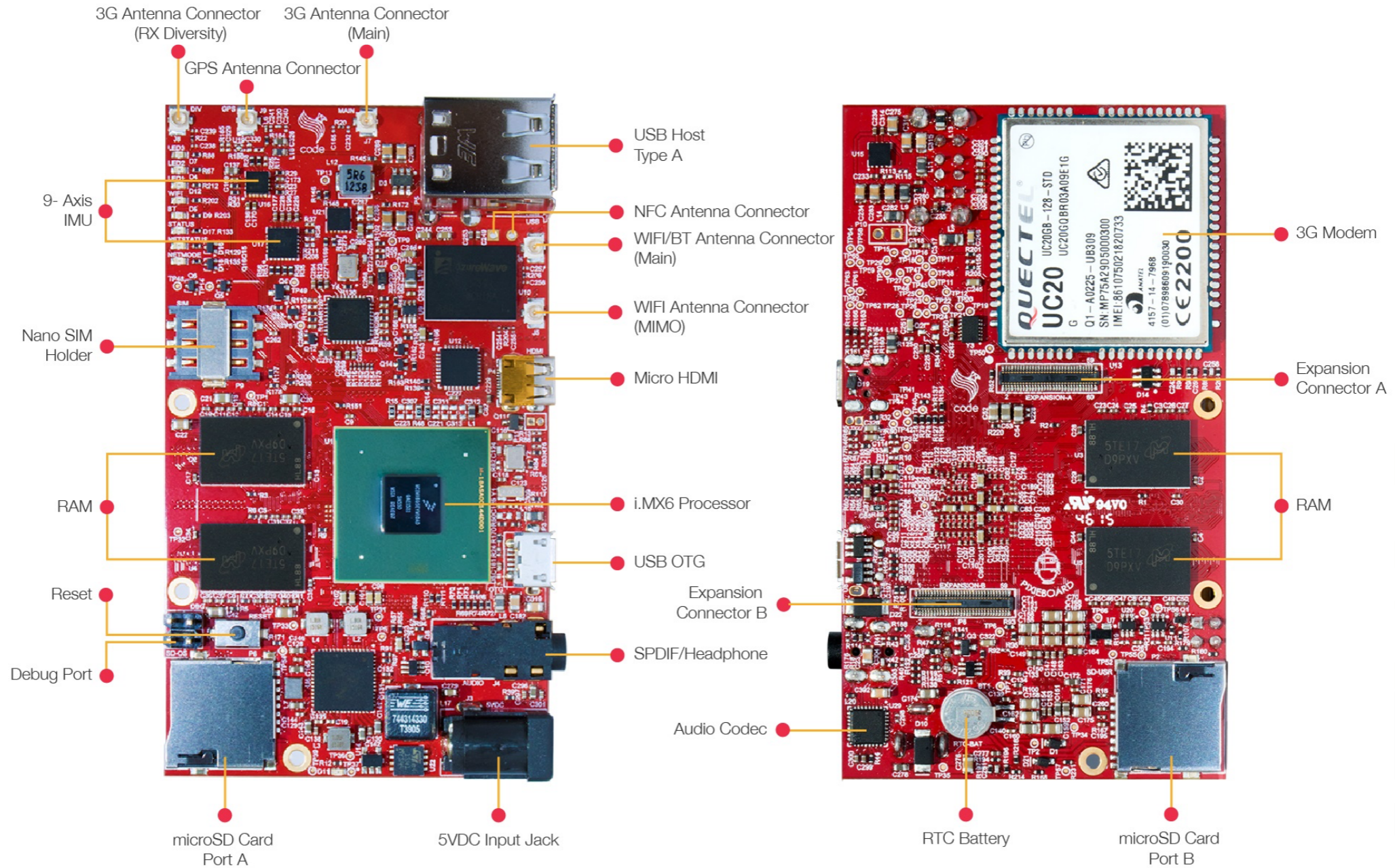
The idea behind PixiePro

- Current professional development tools are not ideal for a quick PoC/Demo
 - Software is often very behind the curve
 - Connectivity options are minimal
 - Expensive
 - They are big (they include tons of clunky connectors)
 - Performance is often very different from production hardware





What is PixiePro?

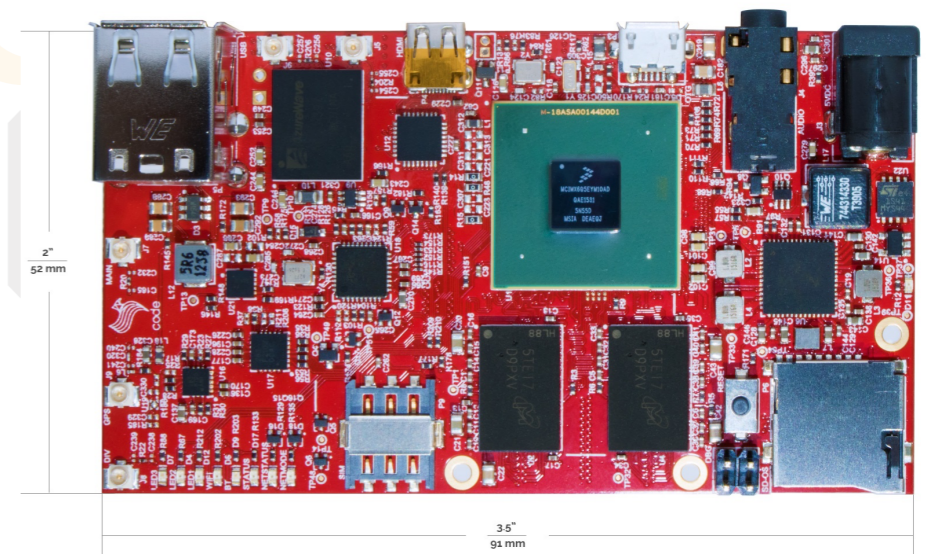


PixiePro is an open source i.MX6 development tool packed with high performance connectivity



Key hardware features

- i.MX6 Quad with 2GB of DDR3
- 2x2 MIMO 802.11 ac WiFi + BT 4.2 + NFC
- Worldmode 3G modem with GPS and GLONASS
- 9-axis IMU (Gyro, Accelerometer, Magnetometer)
- Expansion port
 - PCIe, SATA, SDIO, DSI, CSI, RGB, I2C, UART, CAN and GPIO
 - It can be used a SoM
 - Custom variants from 1k units
- Only 52 mm x 91 mm





Specifications

Specs

CPU	i.MX6Q Quad ARM® Cortex™-A9 @ 1GHz
RAM	2 GB 64-bit DDR3
USB	2 Type-A Host USB 2.0 , 1 microUSB OTG
WiFi	802.11 a/b/g/n/ac @ 2.4GHz, 5GHz, 2x2 MIMO
Bluetooth®	4.2 (w/BLE), 2.1 + EDR
NFC	ISO 14443A/B, 18092, 15693, NFCIP-1, NFC Forum
Mobile Broadband	UMTS/HSPA+ 3G module
Location	GPS/GLONASS
Display	micro HDMI up to 1080p @ 60Hz HDTV
Storage	2 UHS-I SD Card Ports with transfer rates up to 104 MB/s
Audio	mini TOSLINK® optical port / 3.5mm line out
Sensors	9-axis IMU (accelerometer, gyro and magnetometer)
Expansion Port	Gigabit Ethernet, PCIe x1, SDIO, 2 MIPI, SATA-II, RGB666, USB 2.0 HS, CAN, SPI, 2 UART, 2 I2C, GPIOs
Multimedia	2D/3D GPUs & h.264 video encode/decode @1080p30
Power Input	5V DC Input
Dimensions	52 mm x 91 mm

4GEEKS



Built to perform

- WiFi chipset is PCIe
 - Faster wireless than most boards wired networking
 - Measured throughput of 700Mbps
- Modem supports UMTS/HSPA+ (14.4Mbps)
 - RX Diversity support
 - LTE optional
- All antenna ports available
- Dual UHS-I microSD ports
- 1Amp USB 2.0 ports

treats
4GEEKS



Software

- Runs ArchLinuxARM
- Kernels 3.14.25, 3.14.52 and 4.1.15
 - Latest u-boot-imx
- Device tree available in NXP i.MX Community repository
- Yocto Layer Available
- PixieOoBeS
- Debian before Q4, Android by the end of Q4





ArchLinuxARM

- Run the latest and greatest open source packages
- Drivers are ready to use from Linux userspace
 - Network Manager, Modem Manager, gpsd, libnfc, blueman
- Hardware acceleration for popular packages and frameworks
 - Chromium
 - WebGL
 - 2D Canvas
 - GStreamer 1.x
 - Kodi
 - Xorg
 - Qt5
 - Wayland (weston)
 - Cortex A9 optimized versions of
 - gcc, glibc, kernel





Our custom kernel

- Device tree includes support for all the on-board peripherals
- BFS as default scheduler
 - Good balance between real time and throughput
- BFQ as default I/O scheduler
 - Improved performance with eMMC and SD cards
- Source available as a package
- Patches available from the T4G repo
- Supports PixiePro and future PixiePro derivatives
- Building a custom kernel is super-easy using the PKGBUILD source





PixieOoBeS

- Use minimal accessories to start working with PixiePro
 - SD Card
 - WiFi Antenna
 - Power supply
 - USB OTG (Optional)
- SD card resize is automatic
- Configure basic options using a Web Interface
- Access shell using ssh
 - 192.168.12.1 in Windows
 - mightyPixiePro.local in Linux (Avahi) and OS X





PixieOoBeS

The screenshot shows a web browser window with the address bar displaying 'mightypixiepro.local/'. The page header includes the 'Pixie PRO' logo and three navigation tabs: 'Pixie Info', 'Wireless Info', and 'Wi-Fi Manager'. The main content area is divided into three sections:

- Device Information:** A panel with an information icon containing the following text:
 - HOSTNAME: MIGHTYPIXIEPRO
 - OS: ARCH LINUX ARM
 - KERNEL VERSION: 4.1.15-CK2-PIXIEPRO-GA4D2A08
- Services:** A panel with a Wi-Fi icon containing two toggle switches:
 - AUDIO STREAMING (checked)
 - NAS (checked)
- Audio Output:** A panel with a speaker icon containing three radio button options:
 - ANALOG AUDIO (selected)
 - OPTICAL AUDIO
 - HDMI



PixieOoBeS

The screenshot shows a web browser window with the URL `mightypixiepro.local/wireless-info`. The interface features a navigation bar with three tabs: "Pixie Info", "Wireless Info" (which is active), and "Wi-Fi Manager". The main content area is divided into several sections:

- WLP1S0 Interface:** Displays network details for the WLP1S0 interface, including MAC ADDRESS: 80:D2:1D:57:A1:6F, IP ADDRESS: --, and NETMASK: --.
- Modem Settings:** Shows a single line with "--".
- Connected Clients:** A table listing connected devices. One client is shown with MAC address a4:5e:60:e1:d4:03, IP address 192.168.12.170, and hostname Hanzos-MBP.
- Wireless Services:** A list of services with toggle switches: 3G (off), INTERNET SHARING (off), and HOTSPOT (on).
- UAP0 Interface:** Displays network details for the UAP0 interface, including MAC ADDRESS: 80:D2:1D:57:A1:6F, IP ADDRESS: 192.168.12.1, and NETMASK: 255.255.255.0.



PixieOoBeS

The screenshot shows a web browser window with the address bar displaying 'mightypixiepro.local/scan-network'. The page features the 'Pixie PRO' logo and a navigation bar with three tabs: 'Pixie Info', 'Wireless Info', and 'Wi-Fi Manager'. The 'Wi-Fi Manager' tab is active, showing three sections: 'Available Networks' with radio buttons for 'W Austin' and 'travelbot'; 'Hotspot Settings' with a text field for 'HOTSPOT NAME: PIXIEPRO'; and 'Connect to W Austin' with a 'PASSWORD:' label, a text input field, and an 'OK' button.



Development model

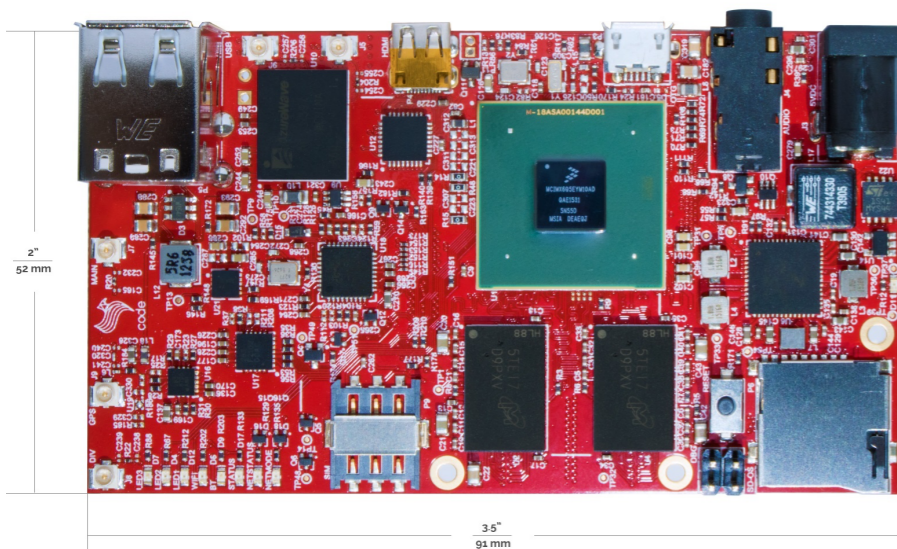
- Download a ready to go image from the Wiki
 - minimal image
 - gui (xorg) image
- Develop directly on PixiePro
 - Support for python, ruby, go, php, node, java, c, c++, etc
 - Transfer files quickly and easily using high speed WiFi
 - Configure hardware and networking from the GUI



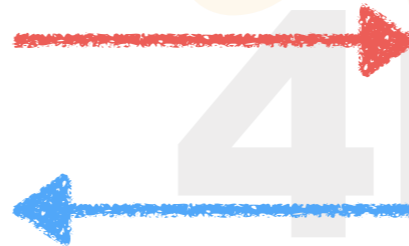


Development model

- Use distcc to compile large projects
 - Use other PixiePros or the VirtualBox (tm) ready-made Virtual machine
 - Reduce compile times in up to 80%
 - Kernel compile under 30 mins
 - Chromium compile under 3 hours



```
main( ) {  
    printf("hello, world");  
}
```



```
300F000068656C6C6F202020202000003C  
311F00007C0802A6900100049421FFF07C6C1B787C8C23783C6000003863000026  
311F001C4BFFFE5398000007D83637880010014382100107C0803A64E800020E9  
3111003848656C6C6F20776F726C642E0A0042  
35030003F9  
39030000FC
```





Expansion boards

- Pinsboard
 - RPi Compatible GPIO and CSI
- Portsboard
 - CAN, Ethernet, GPIO, HDMI, SATA, CSI and RGB
- IoT SubGHz Dongle
 - Compatible with Phalanx Mesh and Sub-Gig 802.15.4





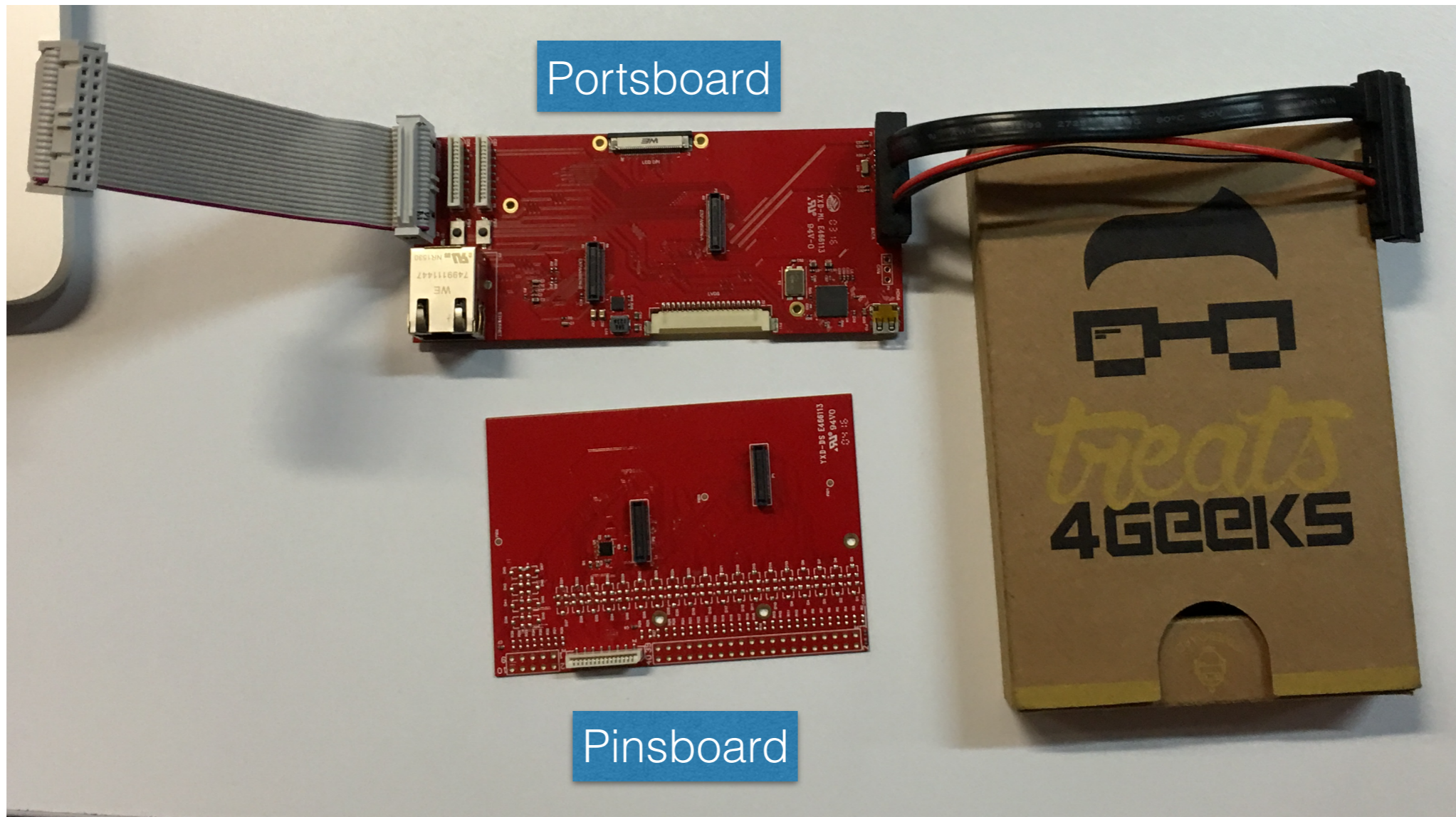
Three easy steps

- Develop your app using the rock solid, ready to go hardware and drivers
- Customize and optimize using the pre-compiled software and frameworks
 - Create your own versions of those frameworks tweaking T4G pkgbuilds
- Create a Yocto minimal image using the Yocto layer if necessary





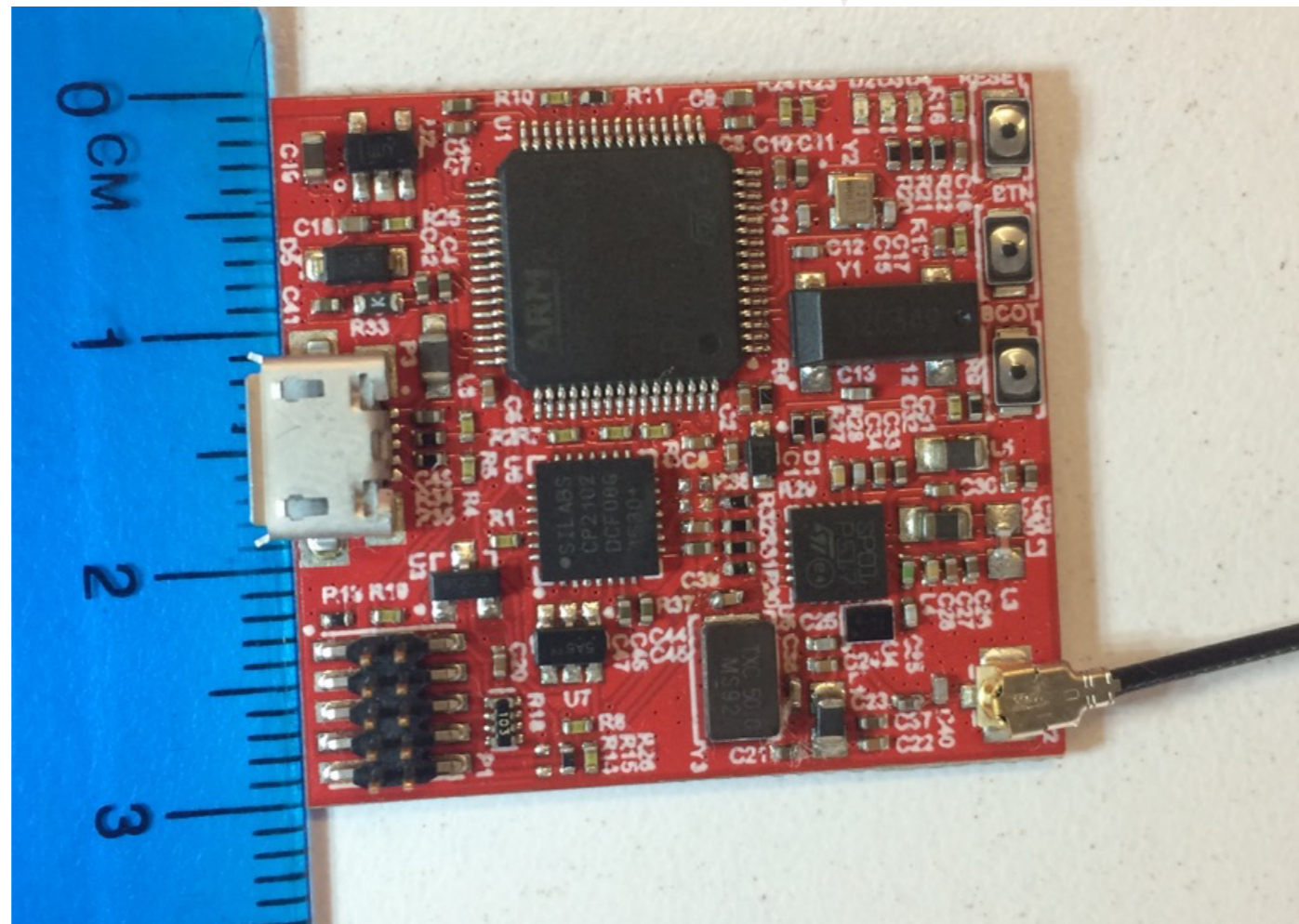
Expansion boards



Portsboard

Pinsboard

IoT SubGHz Dongle



Phalanx IoT Dongle



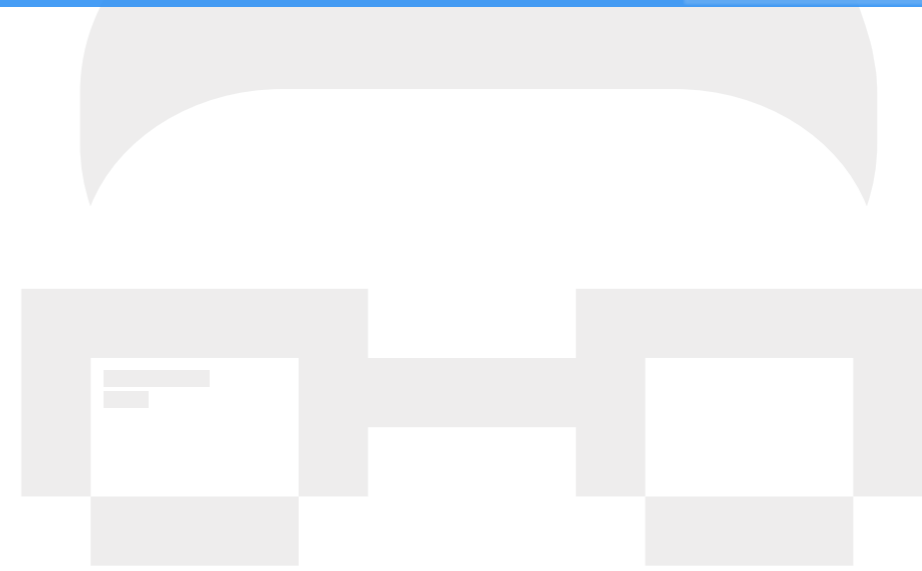
Phalanx

- Rugged IPv6 mesh networking
 - Great in harsh environments
 - Great PER (network management, addressing maintenance and crypto in 16 bytes)
- Tree-mesh
- Highly portable runs from a Cortex-M0+ to an x86 machine
- 5 and 10 node kits available at the store in two weeks
- Licensing model





Face recognition/Azure IoT Hub Demo



treats
4GEEKS