

## Embedded Linux Board Comparison

Created by Tony DiCola



Last updated on 2018-08-22 03:41:12 PM UTC

## Guide Contents

Guide Contents	2
Overview	3
Specifications	3
Input / Output	4
Performance	6
Power Usage	8
Temperature	10
Development	11
Arduino Yun	11
Beaglebone Black	11
Intel Galileo	11
Raspberry Pi	11
Summary	13
Are you new to Linux or programming?	13
Do you need compatibility with Arduino shields or libraries?	13
Do you need to read analog inputs?	13
Do you need real-time control of I/O?	13
Do you need the most performance possible?	13
Do you need WiFi?	13
Do you need to run graphics or media applications?	13
Still not sure?	13

## Overview

---

Have you heard about small Linux-based development boards like the Raspberry Pi or Beaglebone Black, but been confused about which one is best for you? This guide will compare the specifications, performance, power usage, and development experience of four popular Linux-based development boards to help you choose which is right for you!

The boards that will be covered in this guide include:

- [Arduino Yun \(https://adafru.it/d1L\)](https://adafru.it/d1L)
- [Beaglebone Black \(https://adafru.it/tdo\)](https://adafru.it/tdo)
- [Intel Galileo \(https://adafru.it/doY\)](https://adafru.it/doY)
- [Raspberry Pi \(https://adafru.it/aHx\)](https://adafru.it/aHx)





This guide will compare each board with an emphasis on their features and capabilities for maker and electronics projects. All of these boards run a version of Linux and are great for putting sensors, gadgets, and other hardware on the internet. However, you'll see there are quite a few differences in the hardware and capabilities that might make you prefer one board over the other.

**Note:** Since this guide was published an updated Raspberry Pi 2 and Raspberry Pi 3 model were released. See [this excellent comprehensive benchmark of all Raspberry Pi models \(https://adafru.it/tdp\)](https://adafru.it/tdp) for more details on their performance compared to each other and related boards.

## Specifications

---

The following table compares the specs of each board:

	Arduino Yun	Beaglebone Black	Intel Galileo	Raspberry Pi
Picture				
SoC	Atheros AR9331	Texas Instruments AM3358	Intel Quark X1000	Broadcom BCM2835
CPU	MIPS32 24K and ATmega32U4	ARM Cortex-A8	Intel X1000	ARM1176
Architecture	MIPS and AVR	ARMv7	i586	ARMv6
Speed	400mhz (AR9331) and 16mhz (ATmega)	1ghz	400mhz	700mhz
Memory	64MB (AR9331) and 2.5KB (ATmega)	512MB	256MB	256MB (model A) or 512MB (model B)
FPU	None (Software)	Hardware	Hardware	Hardware
GPU	None	PowerVR SGX530	None	Broadcom VideoCore IV
Internal Storage	16MB (AR9331) and 32KB (ATmega)	2GB (rev B) or 4GB (rev C)	8MB	None
External Storage	MicroSD (AR9331)	MicroSD	MicroSD	SD card
Networking	10/100Mbit ethernet and 802.11b/g/n WiFi	10/100Mbit ethernet	10/100Mbit ethernet	None (model A) or 10/100Mbit ethernet (model B)
Power Source	5V from USB micro B connector, or header pin.	5V from USB mini B connector, 2.1mm jack, or header pin.	5V from 2.1mm jack, or header pin.	5V from USB micro B connector, or header pin.
Dimensions	2.7in x 2.1in (68.6mm x 53.3mm)	3.4in x 2.1in (86.4mm x 53.3mm)	4.2in x 2.8in (106.7mm x 71.1mm)	3.4in x 2.2in (85.6mm x 56mm)
Weight	1.4oz (41g)	1.4oz (40g)	1.8oz (50g)	1.6oz (45g)
Approximate Price	\$75	\$55 (rev C), \$45 (rev B)	\$80	\$25 (model A), \$35 (model B)
Documentation	Open source with <a href="#">full schematics</a> . CPU not officially documented.	Open source with <a href="#">full schematics</a> . CPU <a href="#">fully documented</a> .	Open source with <a href="#">full schematics</a> . CPU <a href="#">fully documented</a> .	Open source with <a href="#">full schematics</a> . CPU <a href="#">partially documented</a> .

## Input / Output

The following table compares the I/O capabilities of each board:

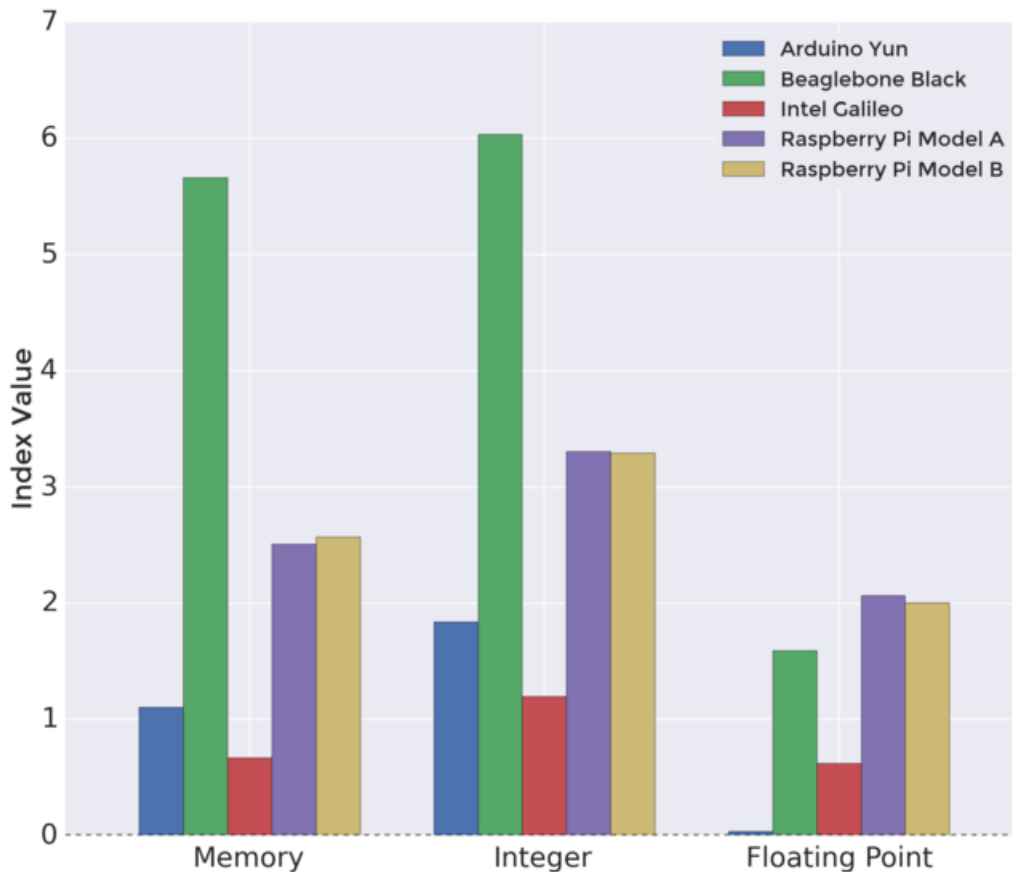
	Arduino Yun	BeagleBone Black	Intel Galileo	Raspberry Pi
Digital I/O Pins	20	65	14	17
Digital I/O Power	5V	3.3V	3.3V or 5V (switched with jumper)	3.3V
Analog Input	12 with 10-bit ADC, 0-5V (supports external reference input)	7 with 12-bit ADC, 0-1.8V (no external reference input)	6 with 12-bit ADC, 0-5V (no external reference input)	None
PWM Output	7	8	6 (limited speeds prevent fine servo control)	1
UART	2 (1 wired to AR9331)	4	2 (1 exposed through 3.5mm jack)	1
SPI	1	2	1	2
I2C	1	2	1	1
USB Host	1 standard A connector (AR9331)	1 standard A connector	1 micro AB connector	1 (Model A) or 2 (Model B) standard A connector
USB Client	1 micro B connector (ATmega)	1 mini B connector	1 micro B connector	None
Video Output	None	Micro HDMI	None	HDMI, Composite RCA, DSI
Video Input	None	None	None	CSI (camera)
Audio Output	None	Micro HDMI	None	HDMI, 3.5mm jack
Power Output	3.3V up to 50mA, 5V	3.3V up to 250mA, 5V up to 1A	3.3V up to 800mA, 5V up to 800mA	3.3V up to 50mA, 5V up to 300-500mA
Other	<ul style="list-style-type: none"> <li>- All I/O routed to ATmega processor unless noted otherwise.</li> <li>- Hardware compatibility with most Arduino Leonardo compatible shields.</li> </ul>	<ul style="list-style-type: none"> <li>- Real-time support with programmable real-time units.</li> <li>- Many pins have multiple functions such as I2S audio, CAN bus, etc. <a href="#">See the wiki</a> for more information.</li> </ul>	<ul style="list-style-type: none"> <li>- Mini-PCI Express slot.</li> <li>- Real-time clock with optional battery.</li> <li>- Mixed compatibility with Arduino shields.</li> </ul>	

## Performance

To compare performance of the boards I used the [nbench](https://adafru.it/doZ) (<https://adafru.it/doZ>) benchmark tool. This is an old tool that was originally created to measure the performance of early Pentium-class computers. When nbench runs it performs a series of tests which are meant to mimic real world workloads, like compressing data or training a neural network. Each test result is combined to build a score for the memory, integer, and floating point performance of each system.

For this test I compiled nbench for each board using GCC 4.7. I chose a minimum of optimization settings as the goal was to compare each board and not necessarily produce the fastest results.

### nbench Results



	Arduino Yun	Beaglebone Black	Intel Galileo	Raspberry Pi Model A	Raspberry Pi Model B
Memory Index	1.104	5.661	0.669	2.509	2.570
Integer Index	1.840	6.032	1.198	3.305	3.291
Floating Point Index	0.038	1.591	0.621	2.064	2.002

Looking at the results, the Beaglebone Black has the strongest memory and integer performance. However the Beaglebone Black's floating point performance is slightly behind the Raspberry Pi. This can be explained because the ARM Cortex-A8 processor on the Beaglebone Black has a 'VFPLite' floating point unit which isn't as fast as other ARM FPUs. If you only care about performance and don't have a floating point heavy workload, the Beaglebone Black is a good board to consider.

Another interesting comparison is the Arduino Yun and Intel Galileo. Both boards run at the same 400mhz clock

speed, but it's apparent the MIPS architecture of the Yun has slightly more performance than the Intel architecture of the Galileo. Floating point performance on the Yun is very low because it doesn't have a hardware floating point unit and must run those operations in software.

Finally, both the Raspberry Pi model A and model B are almost identical in performance. This result is expected because there is no difference between the processor on either board. The model B board only has more memory and peripherals compared to the model A. From a performance per cost standpoint, the model A board at \$25 is quite impressive.

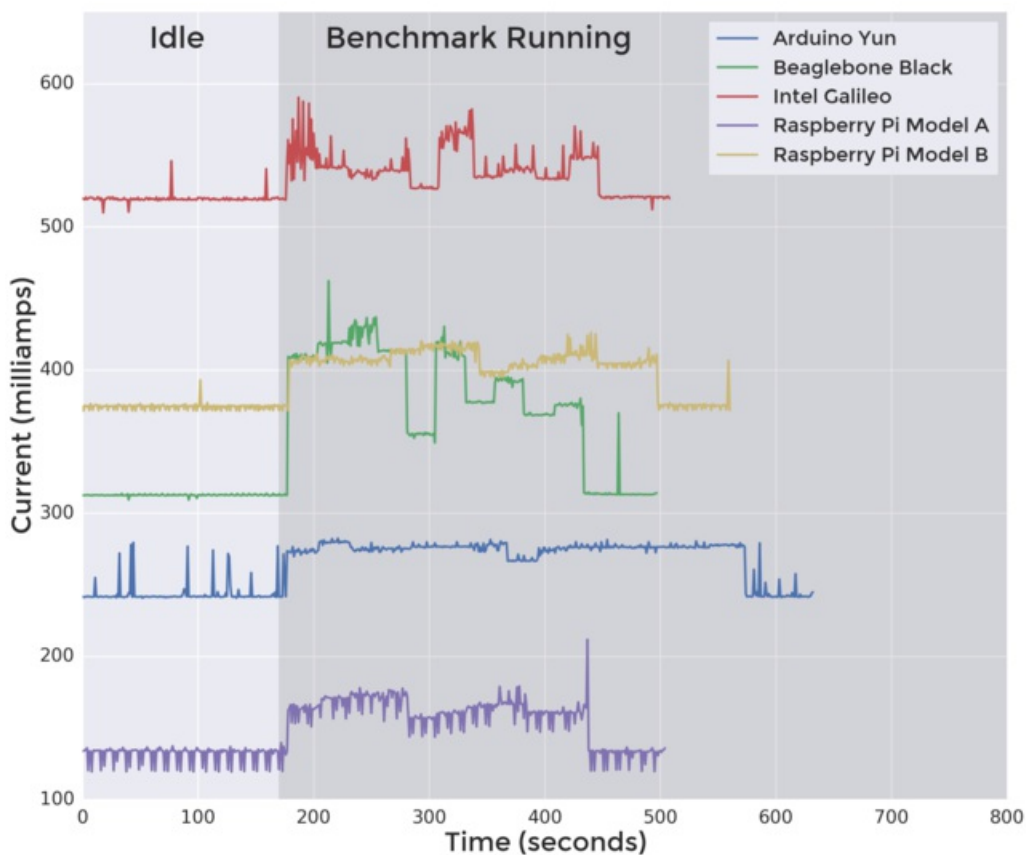
## Power Usage

To compare power usage of the boards I measured the amount of current drawn by each board as it sat idle, and as it ran the nbench benchmark tool. The data from this test can help describe the low and high range for power consumption from each board.

Current usage was measured using an [INA219 breakout](http://adafru.it/904) connected to an Arduino and measuring the current drawn to each board's 5V input (either through a barrel jack or USB micro B connector).

Each board was connected to a minimum of peripherals during the test. The Beaglebone Black, Raspberry Pi model B, and Intel Galileo were connected only to a network through their ethernet port. The Arduino Yun was connected to a network through WiFi. The Raspberry Pi model A was not connected to a network, and instead connected to a USB keyboard and HDMI monitor.

### Board Power Usage



From the results above it's interesting to see the wide range in power usage. The Raspberry Pi model A is at the low end with ~150mA of average current draw. At the high end the Intel Galileo board consumes well over 500mA of current, even at idle! This large difference is likely a result of the difference in peripherals and supporting chips on each board. The Galileo has a lot of peripheral chips like an I/O extender, analog to digital converter, ethernet adapter, etc. which all consume power, whereas the model A board is just the Broadcom processor and very little else.

Both the Beaglebone Black and Raspberry Pi model B have similar power usage under load, but the Beaglebone Black is noticeably lower at idle. Even with its greater performance in the benchmark, the Beaglebone Black is slightly better at power usage than the Raspberry Pi model B.



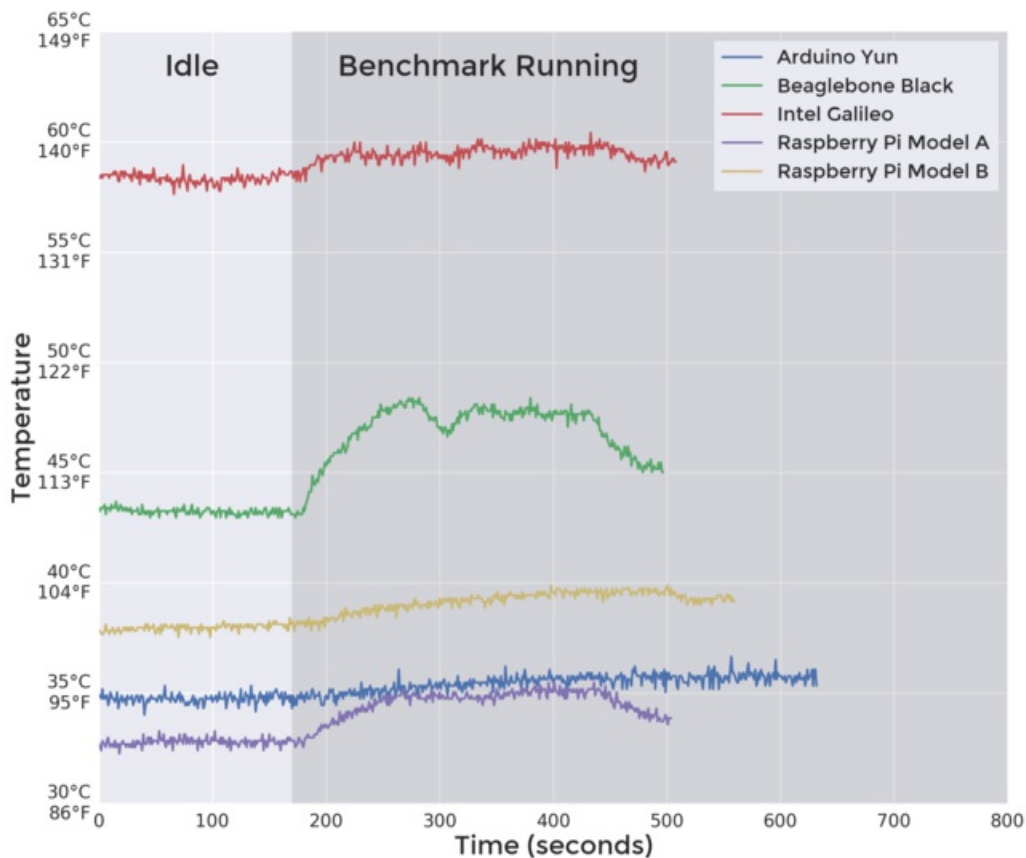
Finally, it's interesting to see the impact of WiFi with the Arduino Yun results. While running over WiFi the Yun demonstrated low power usage compared to the other boards. To see how running other boards with WiFi would compare, I ran a quick test to measure the power consumption of the Raspberry Pi model A board at idle and under load with a [small USB WiFi adapter \(http://adafru.it/814\)](http://adafru.it/814) connected. Power usage with the WiFi adapter increased by about 30-150mA, depending on network activity, which puts it in line with the Arduino Yun results. In general running a board with WiFi doesn't appear to be a dramatic cost in power usage.

# Temperature

To understand how hot each board can get, I measured the external temperature of each board's CPU while running nbench. Temperature was measured using an LM35D temperature sensor held against each board's CPU with tape. The temperature sensor was connected to an Arduino analog input, and a precision voltage reference was used with the Arduino analog reference input.

Note the Arduino Yun's CPU is covered by a metal RF shield. I chose not to remove this shield and measured temperature of the shield directly above the CPU.

### External CPU Temperature



From the results above you can see the Intel Galileo board runs quite hot, at around 60°C / 140°F. The CPU on the Galileo does not have a heatspreader or large package to absorb heat so higher temperatures like this are likely expected. [According to Intel's datasheet \(https://adafru.it/dp0\)](https://adafru.it/dp0) the Galileo processor can run up to 70°C, which is in line with the data that was measured here. In general be careful of touching or obstructing the CPU on the Galileo since it is very hot to the touch.

Another interesting result is the Beaglebone Black compared to the Raspberry Pi model B. Although the Beaglebone Black has better performance and power consumption, it appears to come with a trade off of noticeably higher temperatures (about 5°C / 10°F higher).

Most of the boards run at or below 40°C / 104°F which is not excessively higher than ambient temperature (about 22°C / 72°F at the time of these tests).

## Development

---

The software and development support of a board is just as important as the hardware. Below is a summary of the available operating systems and development tools for each board.

### Arduino Yun

- Operating system is based on the [OpenWrt \(https://adafru.it/dp1\)](https://adafru.it/dp1) distribution and limited by only 16MB of space available to the entire operating system. Don't expect to find all the tools and utilities in a desktop Linux environment!
- Out of the box Python 2.7 is available for development. Other languages such as PHP or Perl can be installed through the [opkg package manager \(https://adafru.it/dp2\)](https://adafru.it/dp2).
- Unfortunately there is not enough space to run [GCC \(https://adafru.it/cPp\)](https://adafru.it/cPp) and compile C/C++ code on the board. A [cross-compilation toolchain \(https://adafru.it/dp3\)](https://adafru.it/dp3) must be setup to compile any custom code for the Linux environment.
- All access to the I/O is done exclusively through the ATmega32U4 processor. Programming the ATmega processor is simple with the Arduino IDE, much like programming an Arduino Uno/Leonardo.
- Communication between the Linux environment and ATmega processor must be done through a serial connection. The [Bridge library \(https://adafru.it/d8L\)](https://adafru.it/d8L) provides a way to communicate and send tasks from the ATmega to the Linux environment.
- Real-time control of devices is easy by writing code to run on the ATmega processor. Many of the shields, devices, and libraries that are compatible with the Arduino Uno/Leonardo are compatible with the Yun.

### Beaglebone Black

- Great operating system support with the availability of [Debian, Ubuntu, Android, and many other operating systems \(https://adafru.it/tdq\)](https://adafru.it/tdq) for the board.
- Installation of compilers and programming languages (such as GCC, Python, Ruby, Node.js, Perl, etc.) is very easy from packages in the supported Linux distributions.
- Out of the box provides a web-based [Cloud9 IDE \(https://adafru.it/dp4\)](https://adafru.it/dp4) and [Bonescript \(https://adafru.it/dp5\)](https://adafru.it/dp5) (Node.js and Javascript-based) interface to I/O.
- Huge amount of I/O available to the board, however access to the I/O is complicated by [manipulating the device tree in Linux \(https://adafru.it/dp6\)](https://adafru.it/dp6).
- Real-time control of devices is possible with the [programmable real-time units \(https://adafru.it/dp7\)](https://adafru.it/dp7), however there's not a lot of tools or libraries to make development with them easy yet.

### Intel Galileo

- Operating system is a custom build of Linux created with [Yocto \(https://adafru.it/dp8\)](https://adafru.it/dp8). Software support is quite limited out of the box; a 'full' distribution of Linux must be installed on a micro SD card.
- Development tool support is also very limited out of the box. A [3rd party build \(https://adafru.it/dp9\)](https://adafru.it/dp9) must be installed to get access to GCC and other development tools.
- Modified Arduino IDE provides primary interface for programming the I/O of the board.
- Compatibility with Arduino shields and libraries is limited. Access to I/O is orders of magnitude slower (and not necessarily real-time) compared to a real Arduino.

### Raspberry Pi

- Like the Beaglebone Black, operating system support is great with options like [Raspbian \(https://adafru.it/d6i\)](https://adafru.it/d6i),

[Occidentalis \(https://adafru.it/dpa\)](https://adafru.it/dpa), and [more \(https://adafru.it/dpb\)](https://adafru.it/dpb).

- Installation of compilers and programming languages (such as GCC, Python, Ruby, Node.js, Perl, etc.) is very easy from packages in the supported Linux distributions.
- Access to I/O is easy with support from [libraries in many programming languages \(https://adafru.it/cbG\)](https://adafru.it/cbG).
- No real-time support so interfacing with hardware that has strict timing requirements (like NeoPixels!) is not possible directly.
- Largest [support community \(https://adafru.it/dpc\)](https://adafru.it/dpc) of all the boards, with many [tutorials and guides \(https://adafru.it/dpd\)](https://adafru.it/dpd) available online for learning about the Pi.

## Summary

---

In summary which Linux-based development board is right for you? The answer is, it depends! No single board does everything well, and your needs will dictate which board is best for your project. However, some questions to help guide you to the right board include:

### **Are you new to Linux or programming?**

Stick to a board with a large community of users and good support for programming tools, like the Raspberry Pi. Check out the [Learn Raspberry Pi series](#) here on the learning system to get started.

### **Do you need compatibility with Arduino shields or libraries?**

Look into the Arduino Yun since it has the same processor as the Arduino Leonardo. Be careful of using the Intel Galileo as it has some quirks and incompatibilities with Arduino shields--research what's known to work in their support forums.

### **Do you need to read analog inputs?**

Pick a board with an analog to digital converter like the Beaglebone Black, Arduino Yun, or Intel Galileo. Don't forget you can [add an external ADC](#) to your development board too!

### **Do you need real-time control of I/O?**

Look at a board like the Arduino Yun or Beaglebone Black, which have a small microcontroller built in for running real-time programs.

For other systems like the Raspberry Pi, you might consider adding a [Trinket](#) or other inexpensive microcontroller for real-time I/O control.

### **Do you need the most performance possible?**

The Beaglebone Black showed great performance in benchmarks. The Raspberry Pi is close, and even a little faster with floating point operations.

### **Do you need WiFi?**

The Arduino Yun has WiFi built in, but don't forget a [small USB WiFi adapter](#) can be added to most other boards. The Intel Galileo can even support [PCI express WiFi chips](#).

### **Do you need to run graphics or media applications?**

Although this guide didn't cover graphics, the Raspberry Pi GPU is quite powerful and can [play games like Quake III](#) or watch movies at full 1920x1080P resolution.

### **Still not sure?**

Don't limit yourself to just one board, consider picking up a couple inexpensive boards! A Raspberry Pi model A and an Arduino Uno are a great combination of boards. The Raspberry Pi has a full Linux environment with great performance vs. price, and the Arduino gives you real-time control of a lot of I/O.