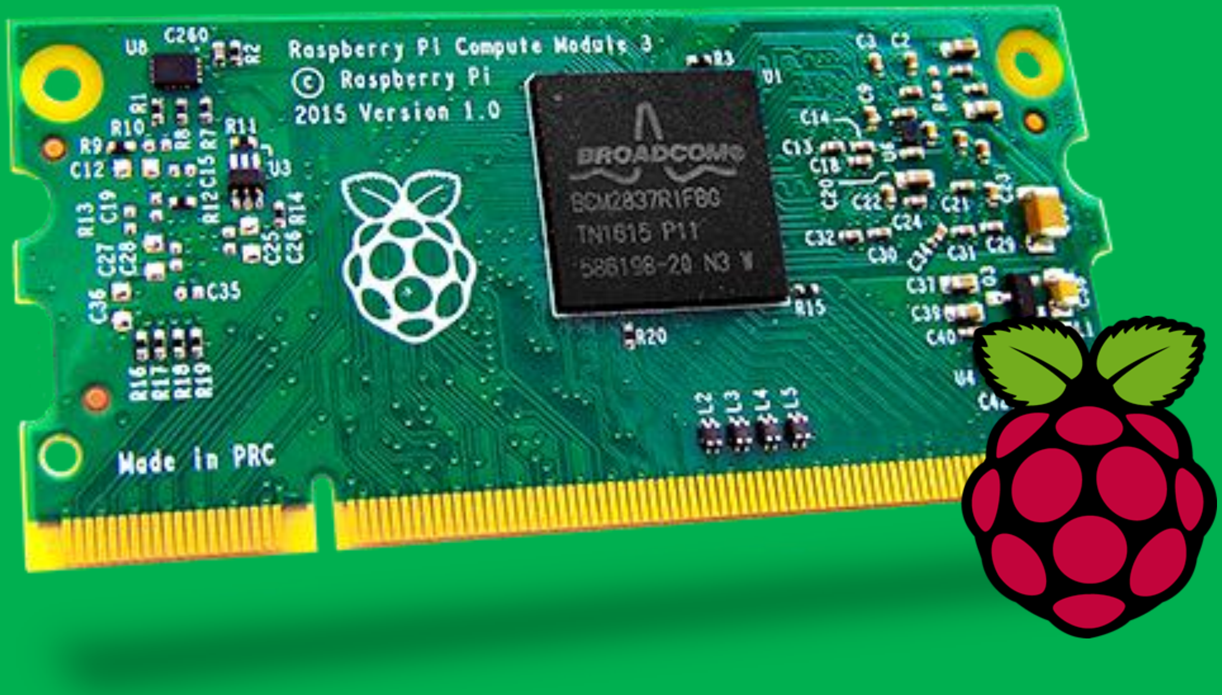
MAKERDEMY

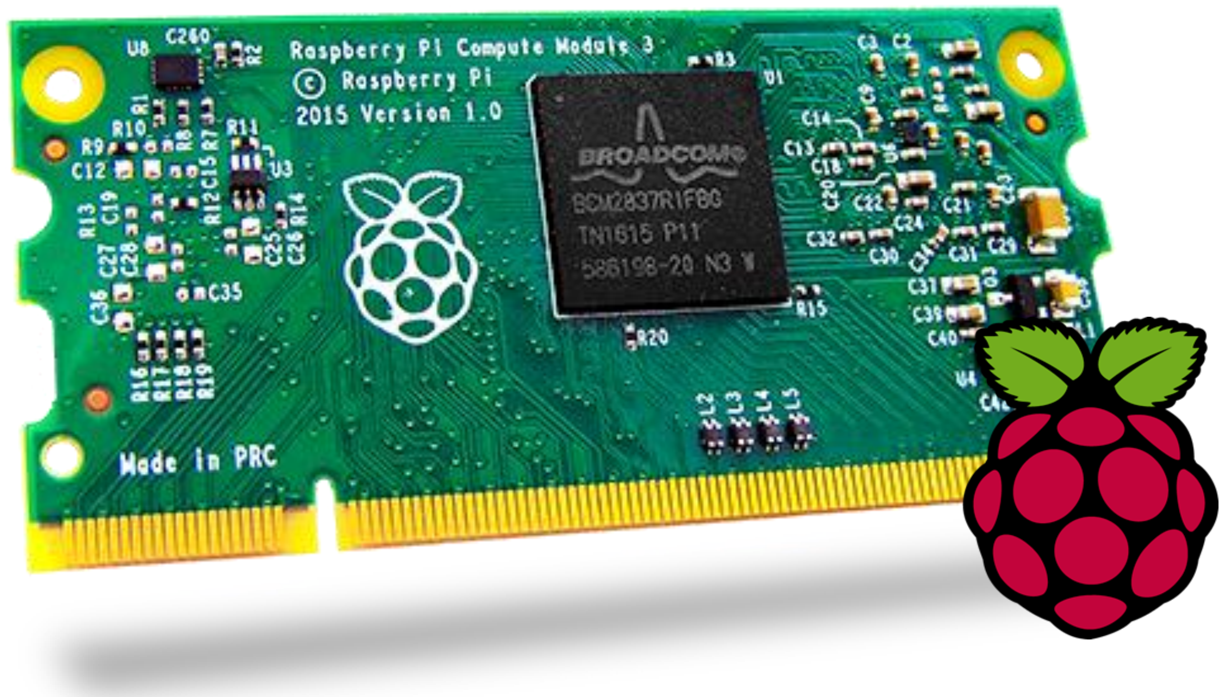
The Developer's Guide to the Compute Module



Rachen Ravichandran

MAKERDEMY

The Developer's Guide to the Compute Module



Rachen Ravichandran

CONTENTS

The 27th LED **1**

Understanding the need for Raspberry Pi Compute Module

Making Sense out of 1, 3, 3L, IO **4**

A deep insight into Compute Module versions - CM1, CM3, CM3L - and its IO board

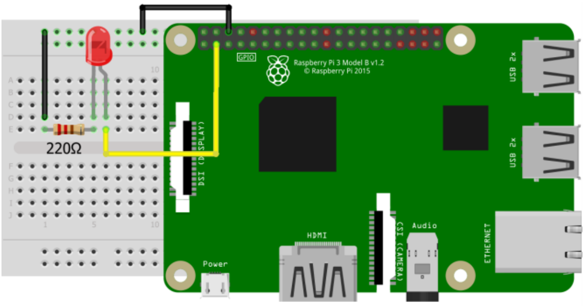
Breaking the CM's code **8**

Decoding the merits, demerits and applications of Compute Module

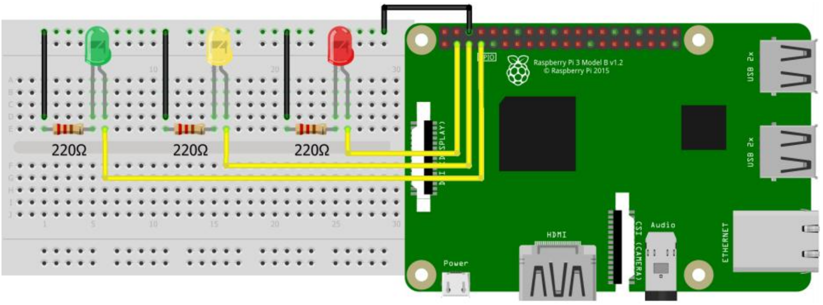
THE 27th LED

Let's do a quick activity.

1. Take a look at your Raspberry Pi 3 (RPi 3.) How many pins do you see? 40? Good. Your vision is definitely clear.
2. Connect an LED to the GPIO pin 2 (refer RPi pinout.) Run the code 'led_1.py' in the RPi. Tada! Your LED blinks!



3. Connect 2 more LEDs to GPIO pins 3 and 4 and run 'led_3.py'. Great!



4. Now, add 26 LEDs to GPIO 2 through 27 and run the code 'led_26.py'.[*] All the 26 LEDs should blink. Child's play.
5. Connect one more LED – the **27th LED** – to the RPi.

But, there are only 26 GPIO[†] pins in RPi 3. Then, where would you connect the 27th LED? In general, what if your project needs more than 26 GPIO pins?

[*] You really don't have to do this though as it's only meant for illustrating the need of Compute Module.

[†] There are actually 28 GPIO pins of which two of them are reserved for EEPROM.

For example, if you are planning to build a Home Security System, you might need as many GPIO pins as shown in the table:

Component	Quantity	Location/Remarks	Number of GPIO pins required
16 X 2 LCD screen	2	• Front porch • Hall room	12
4 X 4 Keypad	2	• Front porch • Hall room	16
IR sensors	5 (approx.)	• Door entry • Windows	5
USB camera mounted on Servo Motor	3	Fixed on the exterior walls of the house	3
LEDs	5	For indication	5
		Total	41

Moreover, you might not use all the 4 USB ports, Wi-Fi, Bluetooth, Ethernet, audio jack etc. It's an overkill and a waste of power especially if your system is battery driven. This is not only true for a home security system but also for any embedded system. Generally, developers use only the peripherals required for the product so that it will not be overcrowded by other components, useless to its application.

Although RPi 3 is ubiquitous in hobbyists' environment, it hasn't reached the commercial market yet due to the popular belief that it is designed only for amateurs and hobbyists which is true, in a sense, as it is an educational device to induce children into computer programming.

In addition, if you were to use the BCM2837 processor (or any other) instead of an RPi 3 in your product, you would need to redesign circuitries and write thousands of lines of codes from scratch. That's definitely a huge waste of time.

So, what if you need to use more than 26 GPIO pins, custom design your own board, prototype your model, commercialize your product and look like a pro? But, what if you also want to exploit the RPi 3's programming simplicity, huge community, humongous libraries of choice, powerful specifications and its brain – the Broadcom processor?

Don't panic!

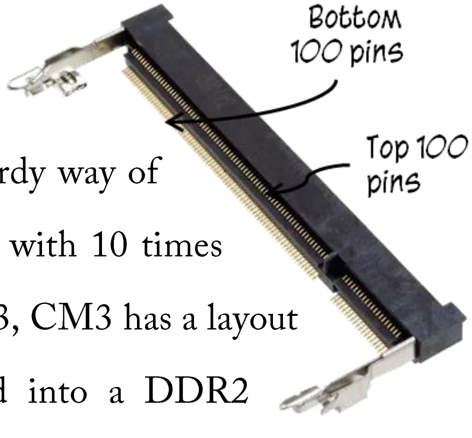
Raspberry Pi Compute Module is the right choice for all your needs. So, what are you waiting for? Go ahead to the next chapter and find out the exciting features this minion offers.

MAKING SENSE OUT OF 1, 3, 3L and IO

Raspberry Pi Compute Module is a prototyping board designed for industrial purposes and comes in three variants:

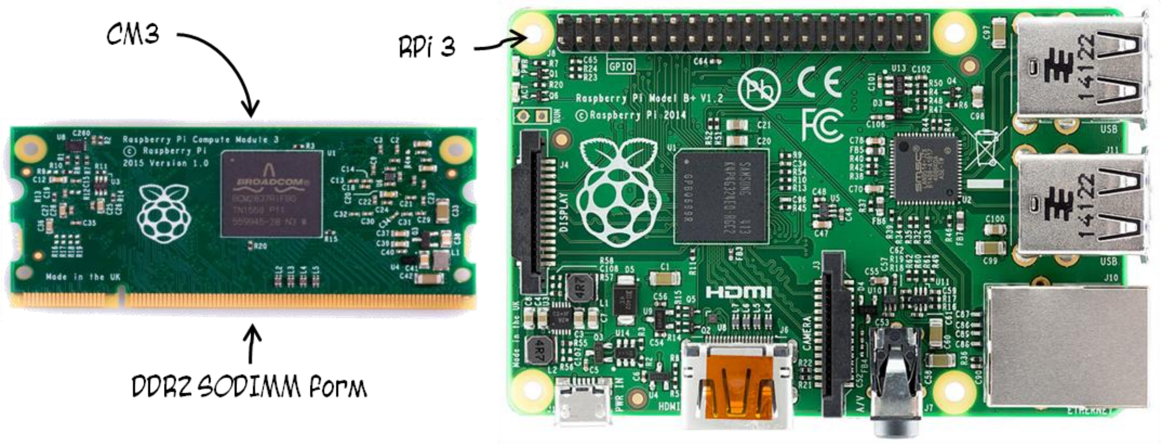
- Compute Module 1(CM1) – April 2014
- Compute Module 3 (CM3) – January 2017
- Compute Module 3 Lite (CM3L) – January 2017

1. CM1 is the oldest of them all which runs on BCM2835 SoC, 512 MB RAM and single core processor with an onboard 4 GB eMMC flash memory (similar to an SD card used in RPi 3.) It is a smaller sized Raspberry Pi Model B+ with inbuilt storage requiring no additional SD card. However, a newer version with a better performance available at the same price, called RPi Compute Module 3, is released. Hence, this version will not be considered further.

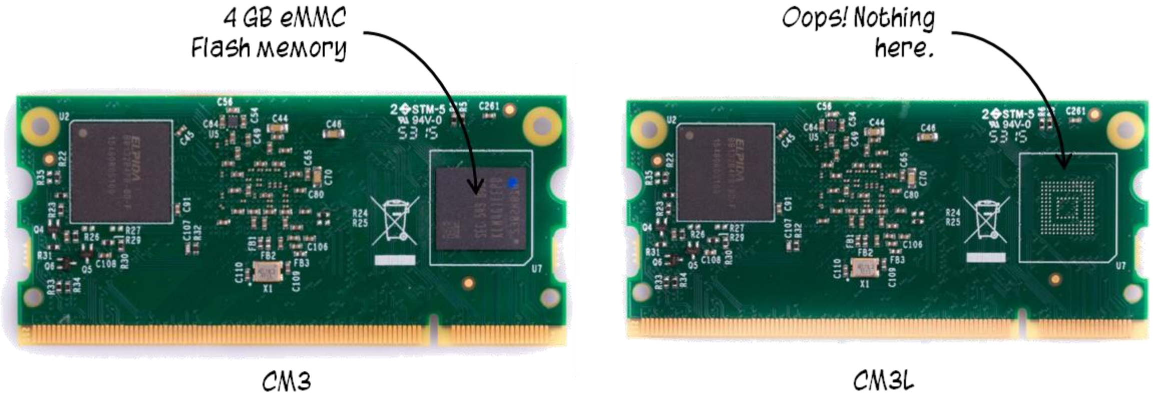


2. CM3 boasts a BCM2837 SoC, quad-core processor, 1 GB RAM and 4 GB eMMC flash memory. It has almost the same form factor (a nerdy way of representing physical size and shape) as CM1 but with 10 times better performance. Unlike the header pins in RPi 3, CM3 has a layout and pins in such a way that it can be inserted into a DDR2 SODIMM connector. This is a type of slot as shown above which can connect to 100 pins at the top and 100 pins at the bottom (totalling to 200 pins.) Designers of CM3 have chosen this form factor as it would be easy to implement the SODIMM slot on the custom board while preserving the small size.

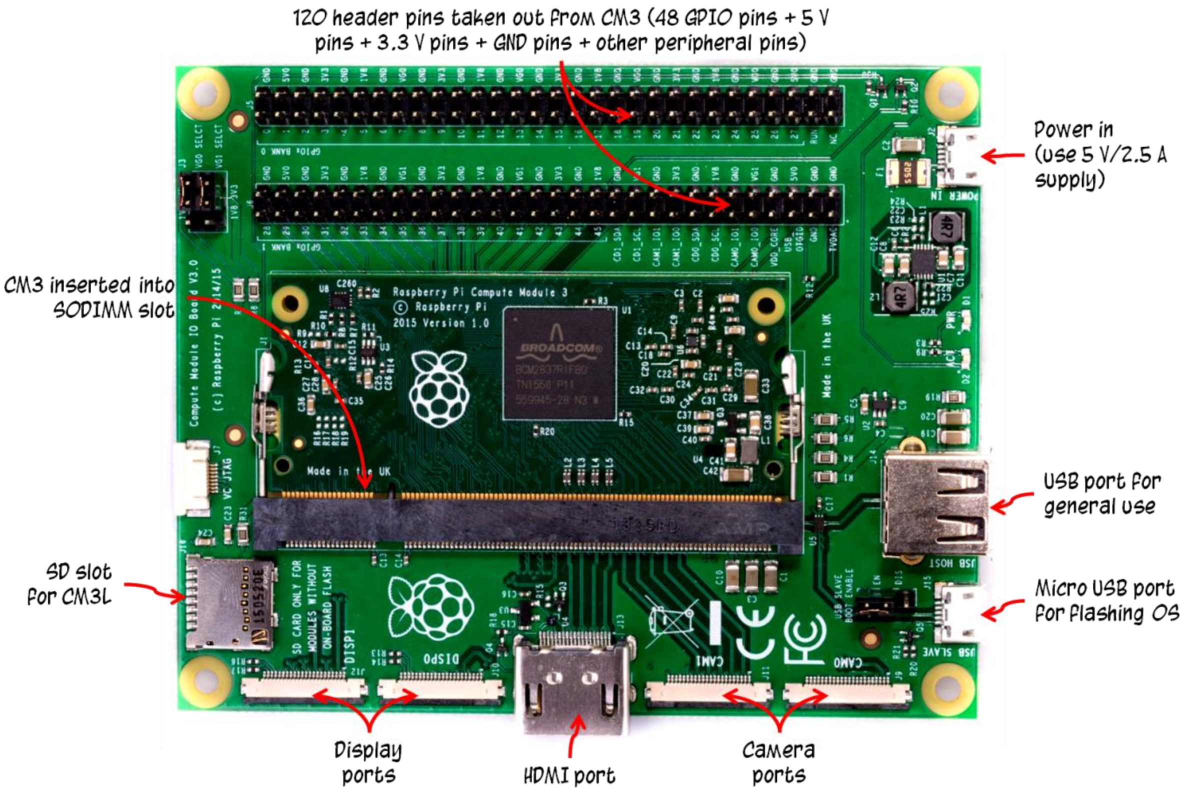
Size comparison between RPi 3 and CM3 is shown below.



3. Compute Module 3 Lite (CM3L) is same as CM3 except lacking the inbuilt 4 GB storage which makes it $5 cheaper. Instead, it uses an SD card for storage which is to be interfaced to CM3L using a few of the 200 pins.



On having a closer look at the module, you'll find that there are no HDMI ports, USB ports or any other ports except for the SODIMM slot. So, one might ask, "What to do with just the Compute Module? Wouldn't it be an ordeal to design a custom board? Where should I even start?" Compute Module IO (CMIO) board, offered by RPi foundation, makes prototyping simple. It is a development board that has a DDR2 SODIMM slot into which you can plug a compute module of any version (shown below).


120 header pins taken out from CM3 (48 GPIO pins + 5 V pins + 3.3 V pins + GND pins + other peripheral pins)
Power in (use 5 V/2.5 A supply)
CM3 inserted into SODIMM slot
USB port for general use
SD slot for CM3L
Micro USB port for flashing OS
Display ports
HDMI port
Camera ports

CMIO

CMIO hosts 120 header pins, 2 CSI (Camera Serial Interface) ports, 2 DSI (Display Serial Interface) ports, an HDMI port, a USB host port and a micro USB slave port for flashing the OS. You can add additional peripherals using the GPIO pins available onboard. It also has an SD card slot to interface your SD card to CM3L. CMIO works similar to an RPi 3 and does not require a custom board.

But, one might further ask, "What distinguishes Compute Module from RPi 3? Is it really worth it to buy one?" Absolutely! For starters, CM has 46 GPIO[‡] pins compared to a mere 26 GPIO pins in RPi 3. Moreover, CM3 has an inbuilt eMMC and a smaller form factor than RPi 3 at a cheaper price.

[‡] There are actually 48 GPIO pins of which two of them are reserved for HDMI hot plug detect and USB boot enable.

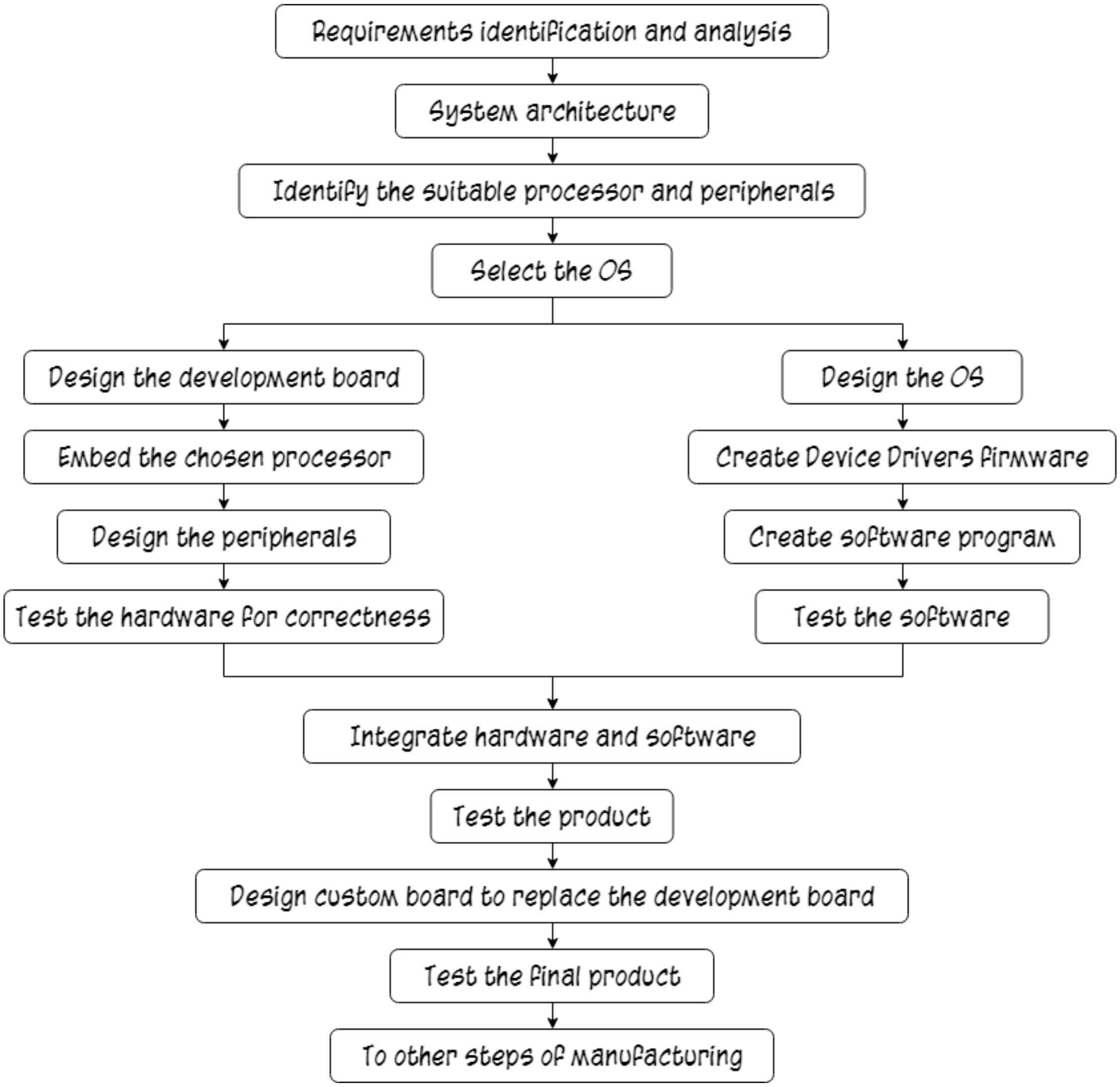
Although Raspberry Pi Zero (RPi 0) has almost the same size as CM3(L) at an extremely low price ($5), the latter has far better performance than the former and allows customized peripherals to be added. For comparison refer the table below.

Model No	RPi 0 (v1.3)	RPi 3 (Model B+)	CM3/CM3L
Price (in $)	5	35	30
GPIO pins	26	26	46
Dimensions	65.0 X 30.0 X 5.0	85.6 X 56.5 X 17.0	67.6 X 31.0 X 5.0
SoC type	BCM2835	BCM2837	BCM2837
RAM	512 MB	1 GB	1 GB
Core Architecture	ARMv6Z (32-bit)	ARMv8-A (64/32-bit)	ARMv8-A (64/32-bit)
CPU Clock	1 GHz Single Core ARM1176JZF-S	1.4 GHz Quad-Core ARM Cortex-A53	1.2 GHz Quad-Core ARM Cortex-A53
Other Functions	2 X I^2C (1 reserved for EEPROM) 2 X SPI 1 X UART 1 X CSI port 1 X DSI (available in GPIO pins) 1 X mini HDMI 1 X micro USB port	2 X I^2C (1 reserved for EEPROM) 2 X SPI 1 X UART 1 X CSI port 1 X DSI port 1 X HDMI 4 X USB port	2 X I^2C 2 X SPI 2 X CSI 2 X DSI 2 X UART 1 X HDMI 1 X USB Host 1 X USB Slave 2 X SD/SDIO 4 GB eMMC (in CM3)

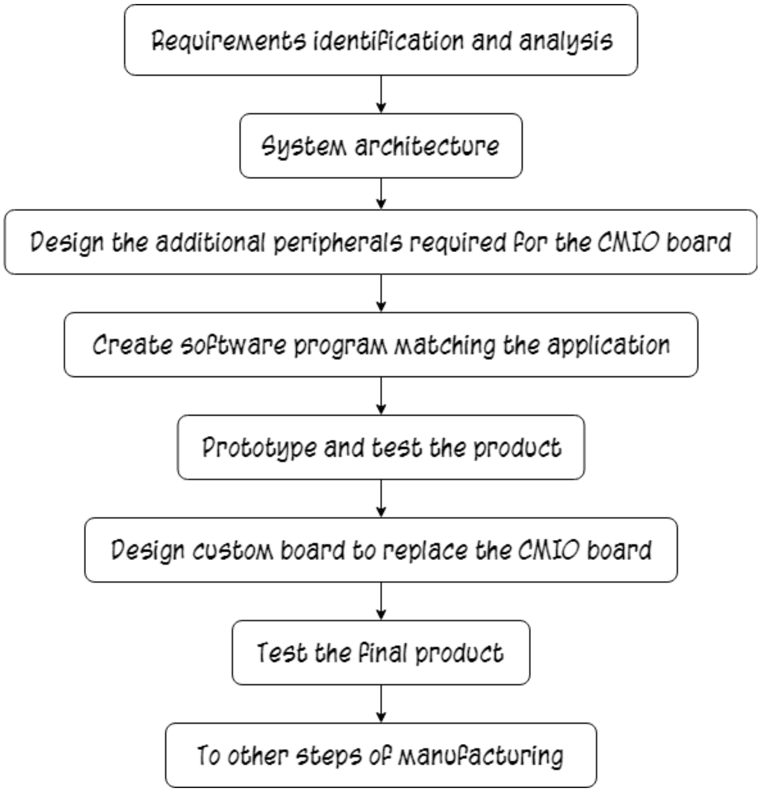
Not convinced yet? CM has many tricks up its sleeves. Find out in the next chapter.

BREAKING THE CM'S CODE

An industrial design process for developing a product is time-consuming and intricate. Without using Compute Module or its IO board, you might be overwhelmed by the extensive design steps and procedures - choosing the right processor and the supporting OS, writing thousands of line of codes for device drivers and software which suits your requirement, design a development board, peripherals and circuitries for the processor, and intensive testing and debugging - during each stage of development.


Requirements identification and analysis
System architecture
Identify the suitable processor and peripherals
Select the OS
Design the development board
Embed the chosen processor
Design the peripherals
Test the hardware for correctness
Design the OS
Create Device Drivers Firmware
Create software program
Test the software
Integrate hardware and software
Test the product
Design custom board to replace the development board
Test the final product
To other steps of manufacturing

However, Compute Module and its IO board allow you to conceptualize, design and prototype your product during its initial stage of development simplifying the design process to a great extent as given below. Once satisfied, you can fabricate your own custom board as per requirement or you can save time by sticking with the CMIO board.


Requirements identification and analysis
System architecture
Design the additional peripherals required for the CMIO board
Create software program matching the application
Prototype and test the product
Design custom board to replace the CMIO board
Test the final product
To other steps of manufacturing

Still confused? Well, you are not the only one to integrate Compute Module into commercial products. There are many developers who've done it already and many who're developing new ones. For instance:

- DIY game consoles created by Freeplaytech^{TM}, a Kickstarter project, embed a CM to play games using RetroPie OS, an OS engineered for retro games.
- An edition of NEC Corp, a display company, incorporates Compute Module at its heart for operation.

- An intern at Argon Design, a UK based company, used 2 cameras interfaced to CSI ports to render a 3-dimensional image for real-time depth perception.

However, there are few drawbacks in integrating Compute Module to the product:

- Linux OS is not suitable for time-critical applications due to the overheads in task scheduling and preemption. An alternative is to run RTOS (Real Time Operating System) and interrupt based system on Compute Module.
- CMIO is more expensive ($115) than CM itself. However, it is just a onetime investment for prototyping. Once the design is complete, developers can begin producing cheaper custom boards of their choice.
- Currently, the manufactured CM boards are less. However, it is expected that this number would rise soon in the near future.
- Some of the open source software restricts commercial use. Developers have to obtain valid license to use them for their product.

In spite of the aforementioned drawbacks, the advantages that the Compute Module offers are far greater than its disadvantages. In a nutshell, if you want to use a larger number of GPIO pins, prototype a custom board integrated with the powerful Broadcom processor at a small form factor, embed it in industrial products, commercialize it, simplify the product's hardware and software design using voluminous libraries and kind-hearted community, and stop reinventing the wheel to save time, the Raspberry Pi Compute Module is the right choice for you.

So, thank me later and get onto that module of yours. Have a happy computing!