Overview of the ARM Embedded Processors from Texas Instruments*

Miguel Morales

This work is produced by The Connexions Project and licensed under the
Creative Commons Attribution License †

Abstract

This module is a brief overview of ARM architectures from Texas Instruments, including a guide on how to select the ARM architecture that is right for your application and a deep dive into the Stellaris ARM Cortex-M architecture.

ARM Embedded Processors – Selecting the right one

ARM embedded processors are ubiquitous. With the vast number of ARM-based processors available, selecting the right one for a senior design can seem daunting. This chapter will help you select the right TI ARM processor for your application.

At a high level, ARM embedded processors can be split into three tiers of performance, the Cortex-A, -R, and –M series.

ARM Cortex-A Microprocessors

- Highest performance microprocessors, use external memory
- For use with a complex operating system
- Applications:
  - Computing, Enterprise, Handset, Digital Home, Industrial, Wireless Infrastructure [ARM website[1]]
- *Author's Note:*

  Use a Cortex-A device if you:

  - Are ready and willing to develop high-complexity software for your application, including running, debugging, and compiling high-level operating system like Windows, Linux, or Android
  - You are using the microprocessor to program a user interface larger than 7" or wish to display any kind of streaming video on any size display
  - You wish to program a field bus protocol

  If you are interested in working with a Cortex-A device from TI, visit the BeagleBone® website here: http://beagleboard.org/bone/[2]

ARM Cortex-R Microcontrollers

*Version 1.1: Sep 3, 2012 5:38 pm -0500

†http://creativecommons.org/licenses/by/3.0/

[1] http://www.arm.com/products/processors/cortex-a/index.php

[2] http://beagleboard.org/bone/

- Medium performance real-time microcontrollers, use integrated memory
- Architected for safety- and life-critical applications with precise real-time requirements
- Applications:
  - Storage, Enterprise, Digital Home, Cameras, Medical, Industrial, Automotive [ARM website[3]]
- *Author's Note:*

  Use a Cortex-R device if:
  - Are ready and willing to develop medium-complexity software, especially if you are interested in safety-critical embedded software development
  - You want to develop a real transportation or safety application

  If you are interested in working with a Cortex-R device from TI, visit the TMS570 Microcontroller USB Kit website here: http://www.ti.com/tool/tmdx570ls20susb[4]

ARM Cortex-M Microcontrollers

- Lowest power microcontrollers, use integrated memory
- Architected for cost and power sensitive applications.
- Applications:
  - Smart metering, human interface, industrial, white goods, consumer, portable medical [ARM website[5]]
- *Author's Note:*

  Use a Cortex-M device if you:
  - Are interested in the lowest-complexity solution for your application
  - Can meet the needs of your application with 150 MHz or less and without a multi-threaded OS running in the system
  - Are looking for the smallest, least expensive, or lowest-power ARM device

  If you are interested in working with a Cortex-M device from TI, visit the Stellaris LaunchPad® website here: http://www.ti.com/stellaris-launchpad[6]

Texas Instruments' (TI) ARM based platforms are outlined in Figure 1, below.

[3] http://www.arm.com/products/processors/cortex-r/index.php
[4] http://www.ti.com/tool/tmdx570ls20susb
[5] http://www.arm.com/products/processors/cortex-a/index.php
[6] http://www.ti.com/stellaris-launchpad

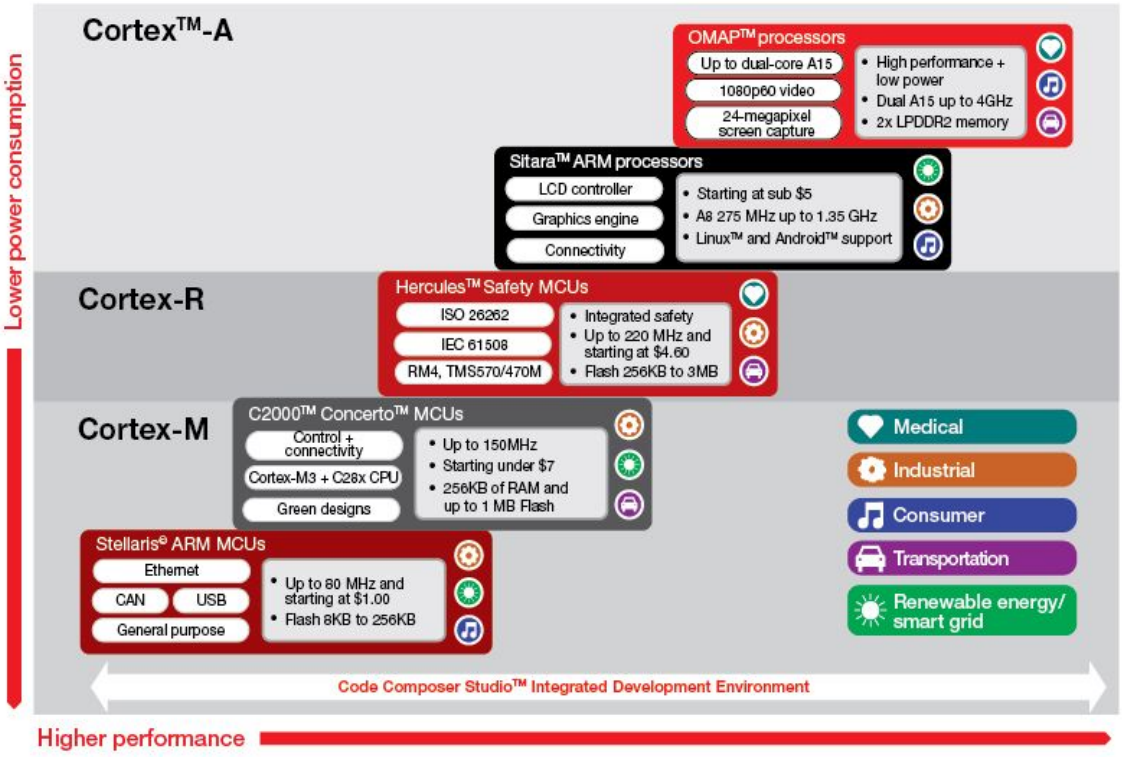

Figure 1: Texas Instruments ARM Processing Portfolio - Aug 2012

Stellaris® Microcontrollers: Offering performance up to 80 MHz and a high degree of connectivity and analog integration to the microcontroller markets, Stellaris MCUs are ideal for applications that require all the memory, analog components, and communications interfaces on a single chip within a compact package.

Concerto™ Microcontrollers: Concerto F28M35x microcontrollers combine TI's performance C28x™ core and control peripherals with an ARM® Cortex™-M3 core and connectivity peripherals to deliver a clearly partitioned architecture that supports real-time control and advanced connectivity in a single MCU.

Hercules™ Microcontrollers: For high-reliability applications such as transportation, the TMS570 provides DSP-like performance with safety-critical features implemented in hardware to ensure continuous and uninterrupted operation. The TMS570 has a dual-core architecture with the second core running in lockstep for redundancy and self-checking.

Sitara™ Microprocessors: Offering performance up to 1.5 GHz, Sitara ARM MPUs are ideal for applications needing a high-level operating system, wired and wireless network connectivity, concurrent applications, and support for rich graphics with an advanced user interface.

OMAP™ Multi-core Microprocessors: The OMAP platforms for digital media processing bring together an ARM core with advanced video acceleration plus a high-performance C6000 DSP for applications that need to support real-time video, image, and audio data processing. With the ability to provide the intensive computational real-time signal processing performance developers can implement complex applications such as video analytics, speech recognition, baseband channel management, and power monitoring with a single chip.

Deep Dive into Cortex-M – Stellaris ARM Cortex-M4F

Stellaris microcontrollers (MCUs) from Texas Instruments are 32-bit, RISC-based, mixed-signal processors designed specifically for ease of use and connectivity. Common applications include end equipments in the compute, industrial, and smart grid markets.

To get a better idea of what Stellaris is and how it can be used to solve an application problem, let's take a look at a typical block diagram for the device. Figure 2 shows the block diagram for the LM4F232H5, one of the LM4F devices:

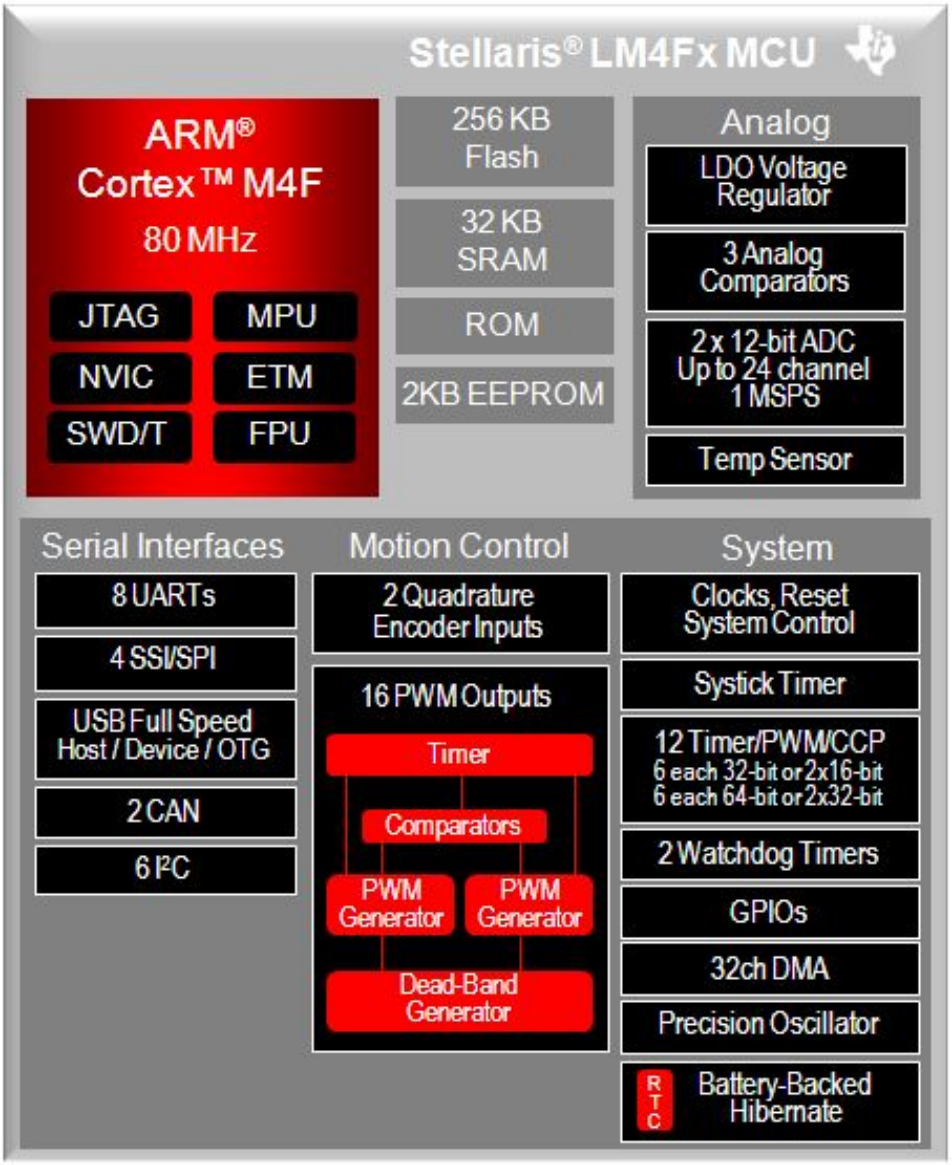

Figure 2: LM4F232H5 Block Diagram

The block diagram provides guidance on the key features of a particular device. Let's take a closer look at some of the more prominent features. Note most of this section is transcribed from the LM4F232H5QC datasheet, available here[7] :

- Integrated Memory
  - Includes both volatile (RAM) and non-volatile (Flash, EEPROM) memories
  - EEPROM supports an endurance of up to 500K writes
  - Read Only Memory (ROM) includes the software libraries used to program peripherals
- Analog
  - ADC: Two 12-bit SAR ADCs, each with up to 12 channels of single- or dual-ended inputs
  - Comparators: Three analog comparators, each with two
- Serial Communications peripherals – both synchronous and asynchronous
  - Two Controller Area Network (CAN) 2.0 A/B controllers
  - USB 2.0 controller, supporting On-the-go, Host, and Device modes
  - Eight Universal Asynchronous Receiver/Transmitter (UART) modules, capable of infrared data (IrDA), 9-bit, and ISO 7816 modes
  - Six Inter-Integrated Circuit (I2C) modules
  - Four Serial Peripheral Interface (SPIs) modules
- Motion Control
  - Two PWM modules, with a total of 16 advanced PWM outputs for motion and energy applications
  - Eight falut inputs to promote low-latency shutdown
  - Two Quadrature Encode Inputs (QEIs)
- GPIOS
  - The 64-LQFP has a maximum 43 GPIOs; the 100-LQFP package has a maximum 69 GPIOs; the 144-LQFP has a maximum 105 GPIOs
  - All pins have configurable pull-up and pull-down resistors
- Timers
  - Systick timer – A 24-bit system timer
  - Twelve general purpose timers - Each 32-bits with up to two capture/compare inputs
  - Two watchdog timers – each 32-bits
- Hibernation Module (HIB)
  - In HIB mode, power supplies are turned off to the main part of the microcontroller and only the Hibernation module circuitry is active. An external wake event or RTC event is required to bring the microcontroller back to run mode.
- Oscillators
  - Precision Internal Oscillator (PIOSC) – on-chip oscillator; operates at 16 MHz ±1% at room temperature
  - Main oscillator (MOSC) – Connection for a single-ended clock source to OSC0 or a high-frequency external crystal [5-25 MHz]
  - Hibernation Module Clock Source – Connection for a 32.758-kHz oscillator
- CPU and Debug capabilities
  - 80 MHz is the maximum CPU clock speed for the LM4F232
  - Other core modules

[7] http://www.ti.com/lit/gpn/lm4f232h5qc

Nested Vector Interrupt Controller (NVIC) – provides programmable interrupt priority and low-latency interrupt handling, including automatic nesting of interrupts
Memory Protection Unit (MPU) – divides the memory map into a number of regions and defines the location, size, access permissions, and memory attributes of each region
Floating Point Unit (FPU) – Supports single-precision add, subtract, multiply, divide, multiply and accumulate, and square root operations. It also provides conversions between fixed-point and floating-point data formats.

- Debug modules

Join Test Action Group (JTAG) port – an IEEE-standard, 4-pin debug interface used to communicate with the LM4F232 during debug
Single Wire Debug / Test (SWD/T) port – an ARM-standard 2-pin debug port, provided as an alternative to JTAG
Embedded Trace Module (ETM) – a real-time trace module providing instruction and data tracing of the Cortex-M4F core
Author's Note – Leveraging the ETM requires the use of a non-standard debugging tool as well as hardware that supports an ETM connection. Though the debugging tools exist, no hardware that supports an ETM connection is available as of September 2012. Go to www.ti.com/stellaris[8] for any updates on the LM4F kits that are available.

Each of these modules is brought out to the user through pins on a package. In the case of the LM4F232H5QC, the Pin Diagram is shown in Figure 3, below. This pin diagram can also be found in the LM4F232H5QC datasheet[9] , under the bookmark called "Pin Diagram".

[8] http://www.ti.com/stellaris
[9] http://www.ti.com/lit/gpn/lm4f232h5qc

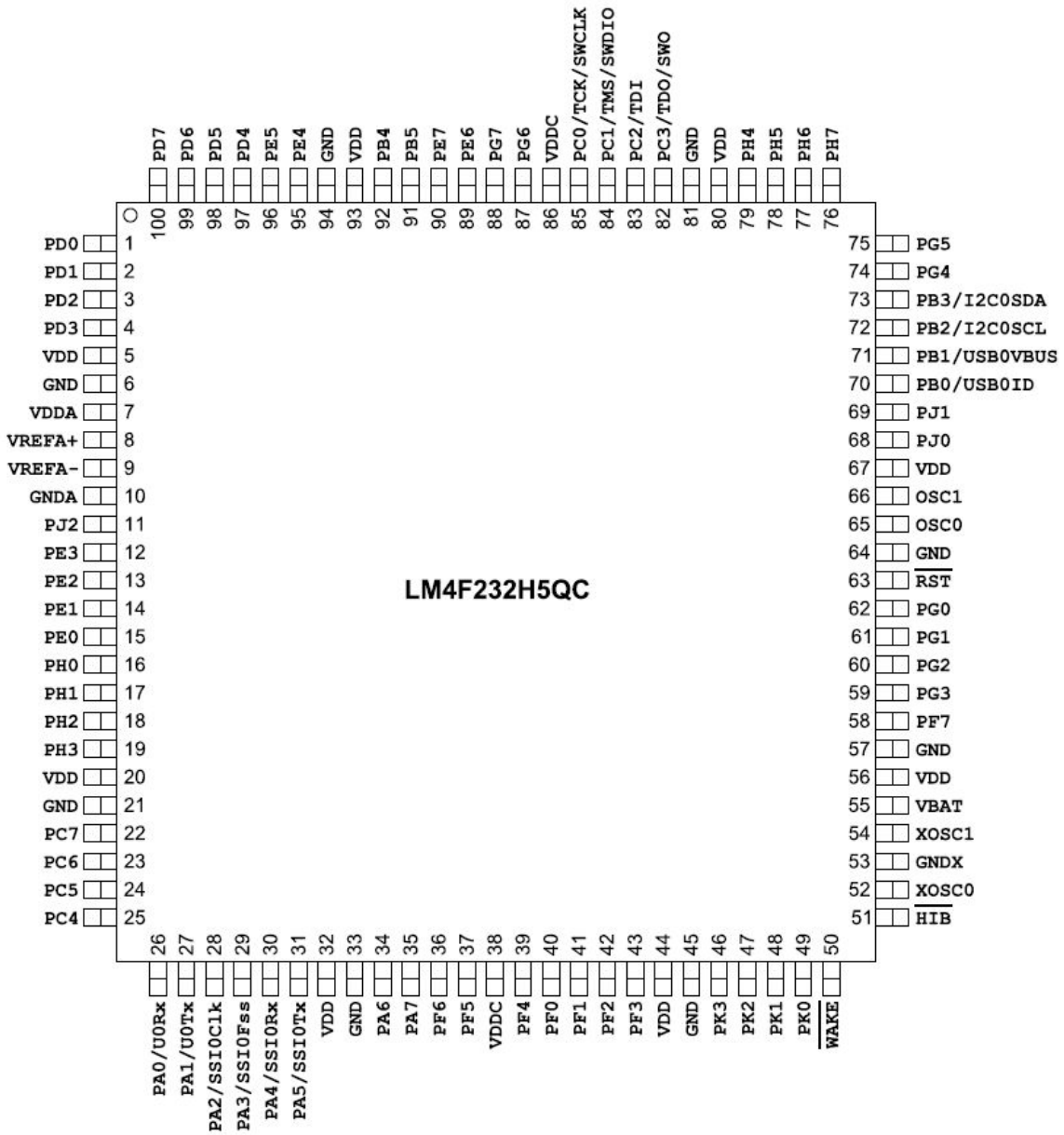

Figure 3: LM4F232H5QC Pin Diagram

This view is helpful because it identifies the necessary connections to the LM4F232, namely the power supply (VDDx, GNDx), the programming/debugging tool connections (TCK, TMS, TDI, TDO), and some of the key peripheral connections (HIB, USB). However, many of the signals such as timer capture/compare pins and analog inputs are still not shown. That granularity is provided in the signal tables in the datasheet. An excerpt from the LM4F232H5QC signal tables is shown in Figure 4.

I - Input
O - Output

Pin Number	Pin Name	Pin Type	Buffer Type[a]	Description
from Pin Diagram	PD0	I/O	TTL	GPIO port D bit 0.
1 Alternate Pin Functions	AIN15	I	Analog	Analog-to-digital converter input 15.
	I2C3SCL	I/O	OD	I^2C module 3 clock. Note that this signal has an active pull-up. The corresponding port pin should not be configured as open drain.
	M0PWM6	O	TTL	Motion Control Module 0 PWM 6. This signal is controlled by Module 0 PWM Generator 3.
	M1PWM0	O	TTL	Motion Control Module 1 PWM 0. This signal is controlled by Module 1 PWM Generator 0.
	SSI1Clk	I/O	TTL	SSI module 1 clock
	SSI3Clk	I/O	TTL	SSI module 3 clock
	WT2CCP0	I/O	TTL	32/64-Bit Wide Timer 2 Capture/Compare/PWM 0.

Figure 4: LM4F232H5QC Signal Table Example

Using these tables, users can start to map the desired functionality to their design. A schematic drawing depicting the baseline connections for an LM4F120 design is shown in Figure 5, below. The LM4F120 and the LM4F232 are actually members of the same device family, so the schematic applies to the part in discussion:

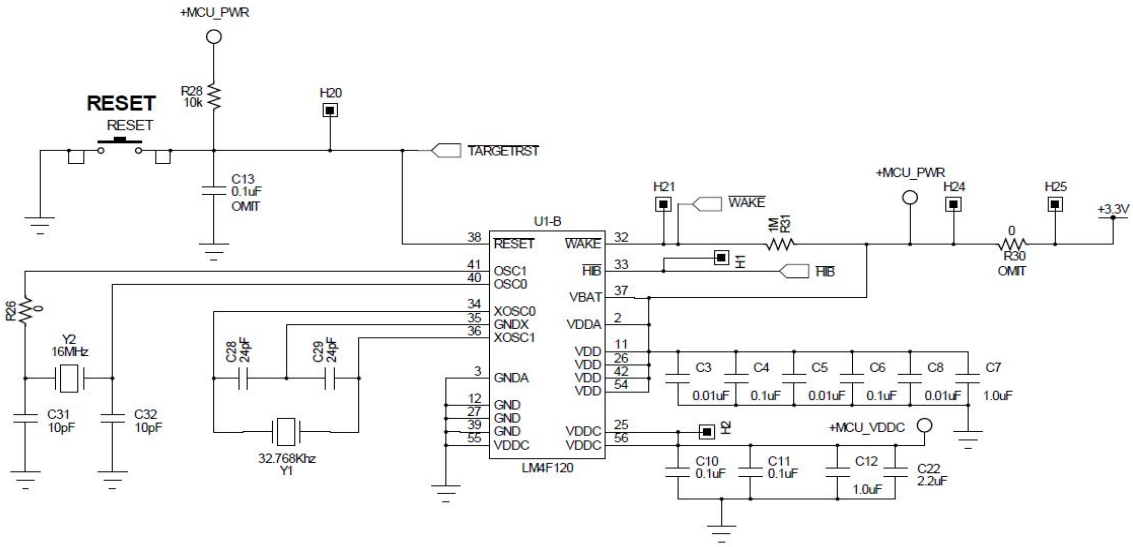

Figure 5: Stellaris LaunchPad (EK-LM4F120XL) Schematic Excerpt

Since the LM4F power rail is specified as 3.3V ± 5%, a typical 3.3V supply will work well. Note the decoupling capacitors on the VDD pins. Signals like the WAKE and nRESET are connected to user switches

that are configured as pull-ups. Two oscillators, a 16MHz and a 32kHz oscillator are connected to the MOSC and HIBOSC pins, respectively.

Figure 6, below, goes one step further to demonstrate an example connection to the debug, USB, and user interface signals.

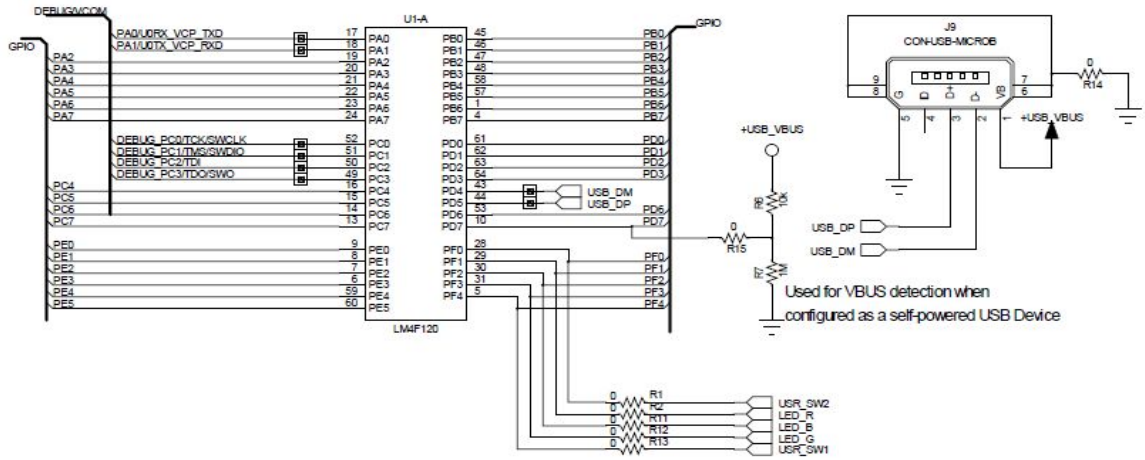

Figure 6: Stellaris LaunchPad (EK-LM4F120XL) Schematic Excerpt

Note the connections to the USB connector and the JTAG signals. Since this is a schematic from the Stellairs LaunchPad[10] evaluation kit, this also includes the connections to a Virtual COM Port bus that uses UART on Port A pins 0 and 1 to communicate back to the debug emulator. The schematic also includes the breakout to the pin headers. These other pin connections can be used to connect all other signals.

Now that we have explained the basics of how to include an LM4F device in your system, let's review the documents you can use to take your design to the next level.

- Device-specific Datasheet (example: LM4F232H5QC Datasheet[11])

  The datasheet includes all the detailed descriptions of the modules, the register descriptions, maximum operating conditions, electrical parameters, performance boundaries, and package descriptions for each device. Each part number has its own datasheet.

- Device-specific Errata (example: LM4F232H5QC Errata[12])

  The errata document is a critical supplement to the datasheet because it lists any bugs and that affect the operation of the device as well as their workarounds. Note that errata vary by revision so it important to map the revision of your device to the errata document.

Datasheet Deep Dive

Beyond the pin diagram and the signal tables, let's highlight some other areas on which to focus when you first look at the datasheet.

[10] https://estore.ti.com/Stellaris-LaunchPad.aspx
[11] http://www.ti.com/litv/pdf/spms318c
[12] http://www.ti.com/litv/pdf/spmz635c

Characteristic	Symbol	Value	Unit
Industrial operating temperature range	T_A	-40 to +85	°C
Unpowered storage temperature range	T_S	-65 to +150	°C

Figure 7: LM4F232H5QC Operating Conditions

Boundary and Recommended Operating Conditions

- Maximum ratings and Recommended Operating Conditions

The maximum ratings indicate the voltage and current conditions that should not be exceeded, under any condition, without risking damage to the device. The recommended operating conditions document the conditions under which the part is guaranteed to operate. This means the minimum and maximum electrical specifications in the datasheet are specified under these conditions.

Parameter	Parameter Name[a]	Value		Unit
		Min	Max	
V_{DD}	V_{DD} supply voltage	0	4	V
V_{DDA}	V_{DDA} supply voltage	0	4	V
V_{BAT}	V_{BAT} battery supply voltage	0	4	V
V_{IN_GPIO}	Input voltage on GPIOs, regardless of whether the microcontroller is powered[bc]	-0.3	5.5	V
	Input voltage for PB0 and PB1 when configured as GPIO	-0.3	V_{DD} + 0.3	V
$I_{GPIOMAX}$	Maximum current per output pin	-	25	mA

Figure 8: LM4F232H5QC Maximum Ratings

Parameter	Parameter Name	Min	Nom	Max	Unit
V_{DD}	V_{DD} supply voltage	2.97	3.3	3.63	V
V_{DDA}	V_{DDA} supply voltage	2.97	3.3	3.63	V
V_{DDC}	V_{DDC} supply voltage	1.08	1.2	1.32	V

Figure 9: LM4F232H5QC Recommended Operating Conditions

Power Consumption numbers
Power Consumption Numbers

Parameter	Parameter Name	Conditions	Nom	Max	Unit
I_{DD_RUN}	Run mode 1 (Flash loop)	V_{DD} = 3.3 V V_{DDA} = 3.3 V Peripherals = All ON System Clock = 80 MHz (with PLL) Temp = 25°C	50	-	mA
	Run mode 1 (SRAM loop)	V_{DD} = 3.3 V V_{DDA} = 3.3 V Peripherals = All ON System Clock = 80 MHz (with PLL) Temp = 25°C	40	-	mA
	Run mode 2 (Flash loop)	V_{DD} = 3.3 V V_{DDA} = 3.3 V Peripherals = All OFF System Clock = 80 MHz (with PLL) Temp = 25°C	30	-	mA
	Run mode 2 (SRAM loop)	V_{DD} = 3.3 V V_{DDA} = 3.3 V Peripherals = All OFF System Clock = 80 MHz (with PLL) Temp = 25°C	20	-	mA

Figure 10: LM4F232H5QC Power Consumption Numbers

- Package information providing package tolerances useful when creating device footprints in schematic/PCB CAD tools:

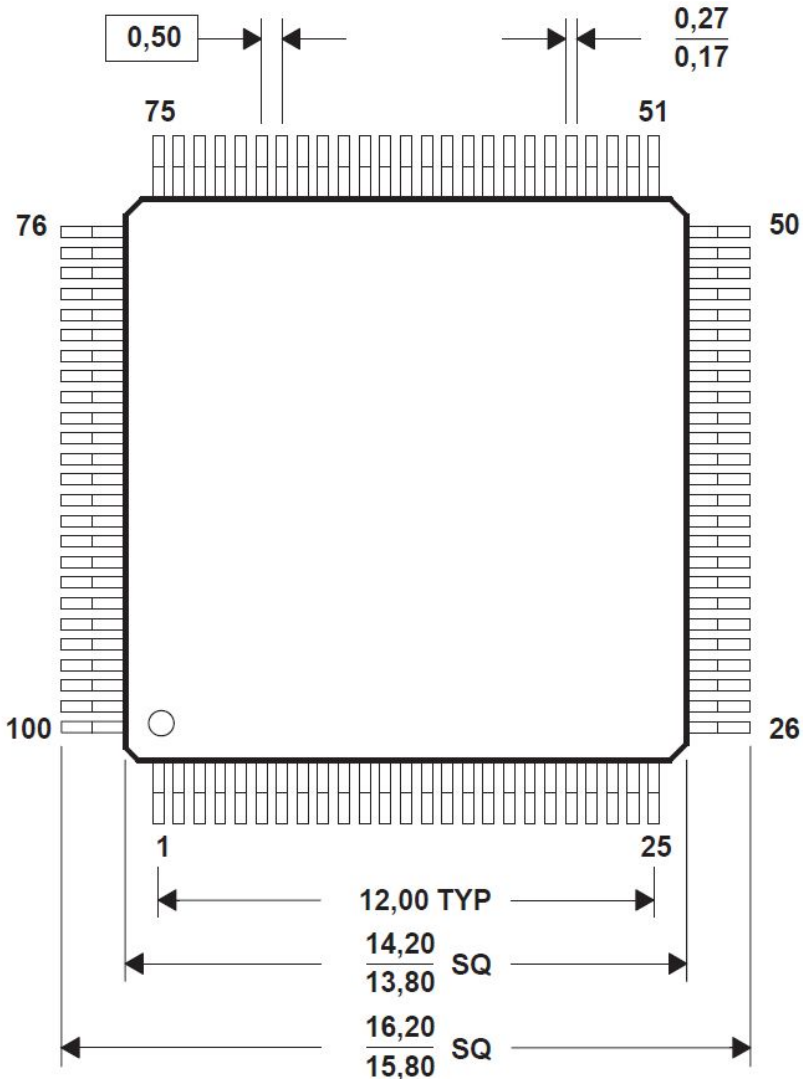

Figure 11: LM4F232H5QC 100-LQFP Package Mechanical Description

The following is an excerpt from "Overview of the MSP430 Microcontroller from Texas Instruments"[13] , written by Zack Albus, changes in brackets "[ ]":

Now that we have reviewed the [Stellaris] device documentation and where to find the needed information about what a device can do, let's walk through the process of actually selecting the part that best suits a given design need.

Let's start by answering the questions below:

1. "What problem am I trying to solve?"

[13] http://cnx.org/content/m44319/latest/

This is fundamental. Until this is understood, nothing else can happen. Knowing what the problem is you face at a system level allows you to identify how features of the MCU can help you solve the problem most efficiently.

1. Do analog signals need to be measured, for instance a voltage from a strain gauge or an output from a potentiometer?

If so, perhaps an integrated ADC is of value. If a simple analog threshold is needed, an integrated comparator can likely do the trick.

1. What is the user interface?

Switches likely translate into simple digital interrupt inputs while displays may require a communication bus such as SPI to be used in order to refresh data displayed to the user.

1. Are time-sensitive signals needed off-chip?

For example perhaps a PWM signal is needed to control a motor's speed or LED brightness.

1. What are the other devices or circuits that the MCU needs to interface with?

Identifying potential analog inputs, logic level digital I/O signals or communication interfaces such as I2C or UART will all help find the right MCU to fit the application need.

1. "How many "things" need to be input into the device or driven from the device (outputs)?"

Determining this at a block level for the system and then at a more detailed MCU pinout level will clarify exactly how many I/O pins are needed

1. "What are my power supply requirements/limitations?"

This is important as the needed of the MCU supply may impact the overall system design. For example if powered from a 9V battery, the [Stellaris] will need a regulator to bring its supply to within the 3.0V-3.6V range.

1. "How much memory will my application need?"

This is not always obvious as the SW may not yet be written. But if you plan on using existing code or modifying code already written, you can get an idea of what a given function in SW might require in terms of program and data memory needs. And when certain algorithms such as FFT or filtering are needed, RAM requirements can often be estimated before ever selecting the device by simulating the functions on a PC.

1. "Are any 'special' features needed?"

For example is a USB interface to a PC needed? Or perhaps a high resolution ADC (>12-bits) is needed to get a certain system performance. Consider looking for MCUs that offer such features integrated to minimize the system-level design effort and complexity required.

1. "Do I have any physical design or assembly constraints? (e.g. package size, pin spacing, PCB or assembly capability)"

[End excerpt]

There are a variety of LM4F devices that will fit a variety of different needs. Sometimes, the easiest way to see whether a device is the right one for your application is to get your hands on an evaluation kit. In the case of the LM4F232 family, the recommended kit is the Stellaris LaunchPad.

The Stellaris LaunchPad is an evaluation kit that costs less than $10 and provides a variety of BoosterPacks to extend its functionality. Make sure to go to www.ti.com/stellaris[14] to start your design today!

LaunchPad

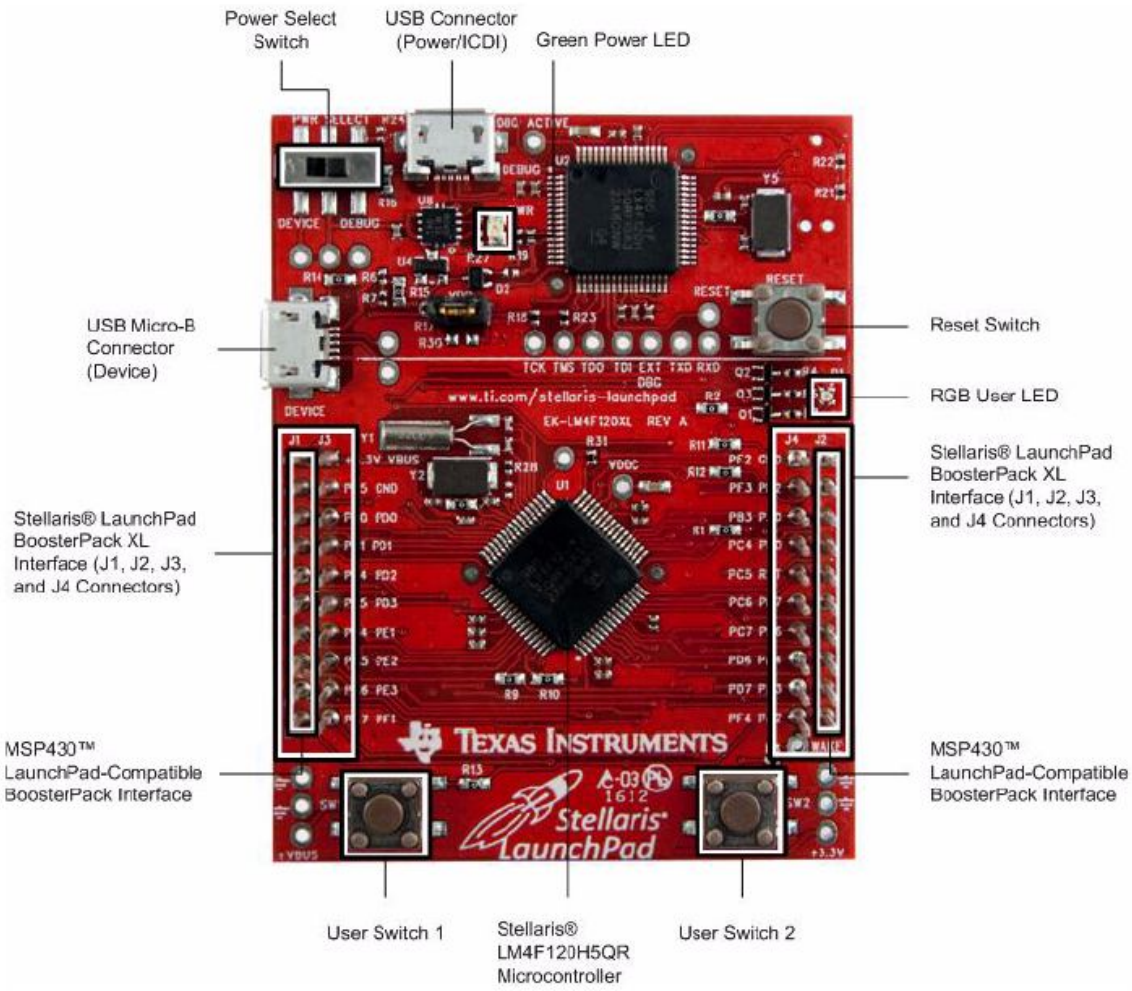

Figure 12

[14] http://www.ti.com/stellaris