

Optimising Sensing Applications with PIC[®] and AVR[®] MCUs

Questions and answers



Question 1

What is the difference between the various numbers after Q, such as PIC18F27Q10 vs PIC18F27Q43?

PIC18F27Q43 – This digit is for the pin count.

- 0 = 14-pin
- 1 = 20 pin
- 2 = 28-pin
- 4 = 40/44 pin
- 5 = 48 pin

PIC18F27Q43 - This digit should stand for the memory variant of the device.

- 4 = 16KB flash
- 5 = 32KB flash
- 6 = 64KB flash
- 7 = 128KB flash

Question 2

Does Microchip offer a solution (as integrated as possible) to convert audio from mic to s/pdif stream?

A suggested starting point is with this device: <https://www.microchip.com/en-us/products/microcontrollers-and-microprocessors/32-bit-mpus/sama7>. It has S/PDIF capability, but you may need external signal chain to get the mic signal into the device.

Question 3

Are you going to add DMA to AVRS in the future? also it seems you are really focusing on MPLAB for your new devices. Is Microchip Studio going to be phased out any time soon?

Microchip Studio will continue to be made available in a frozen state. In that, no new development of the tool including newer device support is planned. Customers are highly encouraged to use MPLAB X IDE.

Question 4

Are those peripherals like OpAmps documented as well as external devices?

Our OpAmp modules have quite a bit of documentation available in datasheets. There are electrical specifications available for many of the important parameters that somebody would need when designing an OpAmp into their application, and for most devices there are charts and graphs available for many of the OpAmp parameters. Additionally, most

of our OpAmps on newer devices have been characterized and included in the MPLAB Mindi simulator which can be used to simulate what the behavior / response would look like.

Question 5

Seems that these MCU are too performant for my app. Does exist more dedicated solutions?

This is a follow on from question 2 and relates to audio processing. Any suggested solution depends on the application requirement. A good starting point is to develop a system block diagram, then consider the requirements for each block. Then, use Mouser's parametric search to identify the product best suited to the system requirements. Microchip has a broad portfolio of products from ASICs, to Power, and Analog signal chain, through to 8-/16-/32-bit MCUs and MPUs, any of which may suit your application. If you're looking for inspiration on where to start with your application, head to Mouser's Technical Resource pages: <https://resources.mouser.com/audio-video>

Question 6

Can you address the supply chain situation for this new device families? - also i would be interested how your projection for older families like the xMega regarding restored availability is?

The newer PIC and AVR devices are manufactured on our latest process node. Since the recent supply chain issues, we have invested in improving manufacturing capabilities, and most of these improvements have been on the latest process nodes. So, the customer can expect the best lead times on these newer devices.

Ideally, we want the customers to move to the newer devices as they not only get the better price but will also be able to take advantage of all the newer peripherals and improvements. The XMEGAs were designed on older process nodes and will continue to be incrementally expensive to manufacture. As Microchip, we will not EOL our devices if a customer wants to buy them, but ideally would like them to move to the newer devices.

Question 7

I am using MPLAB X 5.5, Is there a simple migration for the MCC Project to the new version / Harmony of MPLAB X 6.xx, using newer or optional older version for the code options.

How simple the migration will be depending on the context. Moving your MCC project using the same device should be straightforward assuming that if you created the project using the "MCC Classic" you continue to use that platform instead of the newer "Melody" variant. If you would like to use Melody, then I believe you would need to recreate the project from scratch for the configuration aspect (i.e. setting up peripherals etc....inside of the configurator). Any code created outside of the API should be portable with few edits. I'm not familiar with Harmony projects as I've never personally used.

Question 8

Does the tool chain cover the AT Tiny range?

Depends on which device is being used as to whether it is supported in MCC Melody yet or only currently available in the older version of MCC or if the device is much older than it may not have support in either yet. Inside the MPLAB X IDE, when you open the MCC plugin this information should be presented to you (list of supported devices) if your currently selected device is supported. You won't likely have the option to select a version of the MCC that does not support your selected device. New devices will always be supported first with older devices added as soon as possible. As an example, I loaded up MPLABX 6.10 and was able to create a project for the ATTINY85 and is supported by XC8 compiler but doesn't have MCC support at all...yet.

Question 9

What is the lowest power consumption of all MCUs? Only one OpAmp 4 IO and Serial is needed?

Many of our MCUs offer low-power capabilities which can be used to bring the overall power consumption of the device to less than 100nA (customer should refer to the electrical specs of their device datasheet for the specific number). From there that can be used as a baseline for the lowest power consumption possible, and they could use the electrical specs for OpAmps and serial communication module they want to use to get a good estimate of what they can expect.

Question 10

Is there some official documentation and samples on writing plugins for MPLABX 6.X?

There is a list of SDKs (which match the equivalent MPLAB X version). Latest one is <http://www.opensource4pic.org/content/content/mplab-x-sdk-mplab-x-ide-610>

Question 11

Is there an application note on how to load the firmware from the superior MCU to the PIC?

There isn't a specific application note relating to this. If asking about migrating firmware from say a 32-bit device to an 8-bit device, that suggests migrating from a software centric solution to a hardware centric solution using integrated peripherals. This is very application specific, related to the problem you are trying to solve. A good starting point is to search through Microchip's Discover tool to locate solutions based on market segment, function and even peripherals.

<https://mplab-discover.microchip.com/>