EIU

ELECTRONICS INFORMATION UPDATE

A Mouser Magazine

EUROPE | MARCH 2024

mouser.com

SMART + CONNECTIVITY = EMBEDDED TECH

FEATURES

PLUS

REGULARS

SCAN HERE TO GET THE NEXT EDITION TO YOUR INBOX

Cover image credit: Inkoly/Shutterstock.com

ELECTRONICS INFORMATION UPDATE

mouser.com

In this issue...

We're on the road very soon at the Embedded World razzmatazz in Nuremberg, so to prepare our feature theme this month is Embedded Tech with articles including: Practical security tips; Choosing the right MCU; The evolving edge; Accelerating MPU development; and IoT & battery technology.

We kick off a new series with Qoitech on the low power IoT mindset, and Adam Taylor concludes his machine vision discourse by considering pixels and patterns. Rudy Ramos looks at innovations in e-bike technology, Stuart Cording matches test equipment to your embedded ambitions, and David Pike reminds us that every component impacts reliability. Plus the news round-up, Dev Kit Pick and, of course, a review of the most innovative products now in stock at Mouser. Enjoy!

EIU – Electronics Information Update
Publisher: Mouser Electronics Inc., Georg-Brauchle-Ring 53, 80992 Munich, Germany, represented by Mark Patrick, Director Technical Content EMEA, eu.mouser.com

EIU Editorial Management and Production: Nick Foot, BWW Communications, Suite 6, Festival House, 39 Oxford Street, Newbury, RG14 1JG, United Kingdom, nick.foot@bwwcomms.com, bwwcomms.com

Sales & Marketing: Claudia Bertaccini, Director Marketing Communications EMEA, claudia.bertaccini@mouser.com

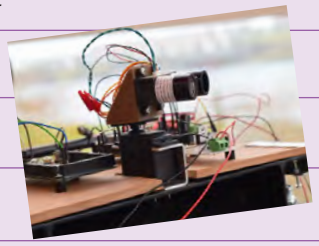
INDUSTRY NEWS

* Renesas to buy Altium * Cars 'to be defined by AI'
* GloFo gets $1.5B CHIPS Act funding * Ingestible thermometer

MOUSER NEWS

* Mouser hits 60 * Soft skills STEM sponsorship * Wear it well

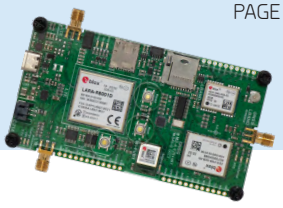
FEATURES



FOCUS

NEW PRODUCTS

Newest products now available from Boundary, Microchip, MultiTech and more.



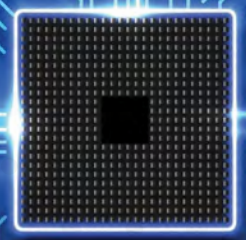


Image credit: Panuwat.ccn/Shutterstock.com

How the AM625SIP Processor Accelerates Development by Integrating LPDDR4

By Mahir Kaheri at Texas Instruments

Selecting the right microcontroller (MCU) is crucial in embedded systems engineering, as it significantly influences the system's performance, power consumption, and overall success.

Processors and microcontrollers are everywhere and used in almost every smart device imaginable. As technology has progressed, end devices and applications have become more sophisticated and smarter to address the needs of our ever-connected world. Indeed, this has also caused processors and embedded systems to become more complicated and larger, leading to increased hardware complexity to address design challenges in applications for smart homes, connected grids factories, and beyond.

This application brief explores common design challenges when designing with a processor. Some of the most common design challenges include:

- Increased hardware and software design time
- Support and robustness of the processor life cycle
- Balancing power consumption with performance needs

Processor Development: Getting to Market Faster

Currently, processors are becoming increasingly larger in size and higher in layer count to address the requirements of new applications with higher performance. For example, a smart home device such as a doorbell camera can require more performance to connect to many accessory devices through local communication and also run processing at the edge to do facial recognition or object detection. A processor in this application can require memory, IOs and significant DMIPs of performance to facilitate these processes.

Ultimately, this can lead to a larger processor which can increase the complexity of the hardware design.

Due to this, there is an increased need for scalability and compatibility between processors. There is also a growing demand for increased computational performance while maintaining compatibility with existing software and hardware. This often leads to more complex tradeoffs and compatibility challenges in processor design when moving from different applications. A doorbell can require 1.4GHz of performance while an Internet of Things Gateway can require less performance.

Instead of redesigning and coming up with a new platform, most designers prefer the scaling of the current processor to several applications. Scalable hardware and software allows ease of reuse of development resources on one processor to another reducing development time and resources in both hardware and software.

Enabling Robustness in Processor Board Design

There are several components including the processor that goes into board design. This includes the processor, memory, peripherals, and many other components. Robustness is a key design consideration in processor selection but extends beyond just hardware and software.

There are additional design challenges in the board design process including security, testing, validation, error handling with booting up board, layout or layer count, and thermal or power management.

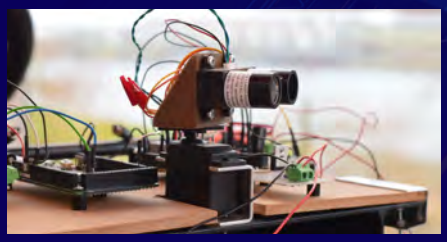
Making sure an end product is reliable, secure, and more resistant to vulnerabilities is crucial. Memory or DDR layout is critical in board design as well as, the memory or DDR layout is the most common reason a board cannot boot up the first time. SoCs need to be able to detect and recover from errors easily.

This is critical but requires extensive testing and validation under various conditions using complex simulation tools. This is not easily feasible to a vast majority of engineers, especially those who are using a processor for the first time.

Successfully meeting the robustness challenge makes sure that an SoC can perform reliably, is more secure, and durable in a wide range of appliances.

Choosing the Right Microcontroller for Your Embedded Project

By Miroslav Milovanovic for Wevolver



Selecting the right microcontroller (MCU) is crucial in embedded systems engineering, as it significantly influences the system's performance, power consumption, and overall success.

Introduction

Embedded systems are integral to our daily lives, found in everything from home appliances to industrial machines. The heart of any such system is its microcontroller (MCU). This tiny yet important component is more than just a piece of hardware; it is the decision-maker that drives the functionality, efficiency, and success of the entire project. Choosing the right MCU is not a mere technical decision; it's a strategic one that significantly impacts the overall system performance, power consumption, and functionality of your embedded design.

A well-chosen MCU ensures that your system runs reliably and efficiently, handling tasks with the necessary speed and consuming power with careful consideration. Conversely, a poorly selected MCU can lead to underperformance, high energy consumption, and increased costs, potentially compromising the project's success. Therefore, understanding the main benefits of MCUs and aligning them with your project's needs is an important part of your project in the field of embedded systems engineering.

Understanding Project Requirements

Selecting the right MCU for an embedded project is an important decision relying on the project's specific requirements, as the chosen MCU directly influences the functionality, efficiency, and scalability of the project. This process begins with a thorough analysis of the project's goals and technical needs. Firstly, it's important to analyze the project's functional requirements, which includes understanding the tasks the MCU needs to perform, such as data processing, signal control, or communication with other devices.

The processing power of the MCU determines how quickly and efficiently it can execute tasks, important for applications that require real-time processing or handling complex algorithms. It is essential to assess the processing needs of your project, whether it involves simple control tasks or more demanding computational work. A more powerful MCU might be necessary for intensive tasks, but it could also mean higher costs and energy consumption.

Memory is another critical factor. The MCU's memory is split into two main types: flash (for storing your program) and RAM (for temporary data storage during operation). The size of your code and the data it needs to handle will dictate the memory requirements. Insufficient memory can lead to performance issues or limit the scope of your project, so it's important to estimate memory needs accurately.

Furthermore, the nature and number of I/O interfaces determine how the MCU will interact with other components like sensors, actuators, and user interfaces. Count the number and type of I/O pins you need and ensure that the MCU can accommodate them. Additionally, consider the need for specialized functions, such as analog-to-digital converters (ADCs) or pulse-width modulation (PWM) outputs.

But also, many embedded projects require MCUs to communicate with other devices. This could be through standard protocols like UART, SPI, or I2C for onboard communication, or more advanced interfaces like Ethernet or Wi-Fi for network connectivity.

The Evolving Edge: Transformation Is Now

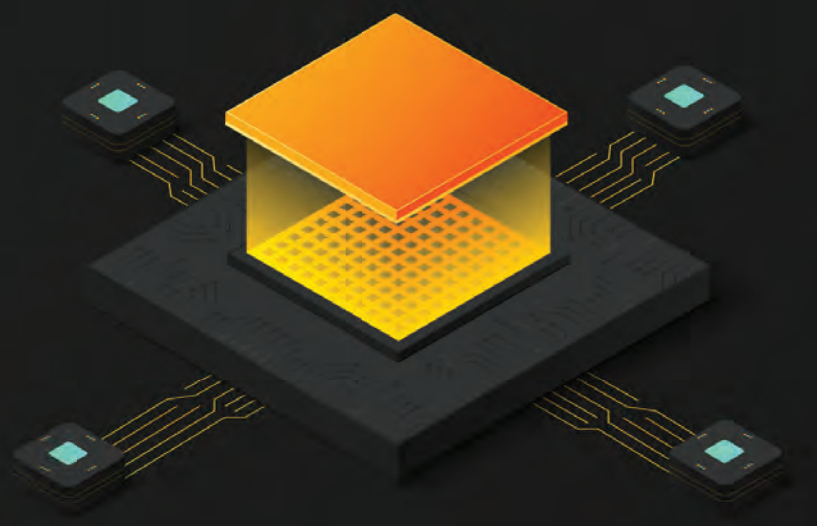


Image credit: Grafvish/Shutterstock.com

By Jon Gabay for Mouser Electronics

The architectures of embedded systems continue to evolve as newer technologies become available to design engineers. What used to be a single central processing unit surrounded by interface and logic circuitry is now a multiprocessor multi-core design with advanced integrated peripheral functionality—many with their own dedicated microcontrollers and resources to contend with.

The once clearly delineated computer edge is now a much vaster and more distributed architecture. Computers that were islands to themselves are now a brick in the wall as they integrate into larger global high-speed connectivity. The hierarchal nature of globally and locally distributed microcontrollers means dedicated processing can occur closer to where data is being generated. The network is the computer.

As higher-resolution data is sensed and processed, chunks of information extraction at each level often result in huge data sets that require a massive pool of immediate and virtual low-latency storage.

Smart houses integrate into a higher level of the distributed processing hierarchy to behave in a city management process domain.

Eventually, the need for our power grid to become a dynamically intelligent entity will be more apparent as electric vehicles and charges become the norm.

In all cases, the Internet of Things (IoT) is central. Everything is connected: from handheld devices to cars and homes, traffic lights to biohazard sensors, factories to cities. This results in each system having multiple edges—some secure, some less so. In this evolving landscape of connectivity, machine learning is crucial as security concerns continually increase with hostile forces trying to wreak havoc.

Machine Learning Is Key

Designers of modern factories, appliances, and more seek machine learning (ML) solutions to tackle new challenges and allow companies to advance and compete. However, low latencies are required to allow useful data integration from extensive and varied sources. As a result, data must be accessed and transported globally very quickly to make these higher-level "thinking machines" respond in an effective and reasonable amount of time.

Image credit: Summit Art Creations/Shutterstock.com

This poses a challenge. Process-intensive security schemes and algorithms can add latency, meaning hardware security acceleration is needed and must be robust enough to protect against threats. Often, vital infrastructure is online and requires high-security protection.

To meet these processing and security requirements, this new generation of designs calls for highly integrated MCUs.

Communications data rates have grown from 50 bits per second (bps) to 10 gigabits per second (Gbps). Software-controlled bit banging will not work at these higher rates, so very high-speed communications hardware must be incorporated into modern microcontrollers.

Additionally, wireless options like Wi-Fi® and Bluetooth®, as well as wired technologies like Ethernet and USB 3.x communications, must peacefully coexist in one system. Intelligent edge devices like facility routers can implement each of these simultaneously. Modern microcontrollers house several encapsulated protocol standards using dedicated accelerated hardware to further unburden CPUs.

Analog and digital signal processing (DSP) requirements must also be addressed, especially with the many IoT devices that include sensor systems as part of their functional requirements.

Image credit: Jackie Niam/Shutterstock.com

Practical Tips for Embedded System Security

By Michael Parks, P.E.
for Mouser Electronics

While embedded systems tend to lack the processing horsepower of servers or even modern personal computers, the sheer number of devices is making them an increasingly valuable target for bad actors looking to run illegal botnets and cryptocurrency mining operations.

One of the first major security-related wake-up calls for embedded system designers was the 2016 Nest thermostat botnet attacks. Given the consumer-facing nature of the particular Internet of Things (IoT) coupled with an increased sensitivity regarding privacy and security; the Nest botnet caused a huge amount of discussion. Those discussions tended to center on how companies should build security into their low-cost IoT products and how consumers can safely operate the devices in their homes and businesses.

With the growing threat of cyberattacks, it is essential that developers keep security considerations in mind throughout the design process. By following some practical tips and recommendations, developers can guard against a wide range of attack scenarios. Read on for an outline of security measures developers can use in their embedded designs.

Build Secure

While there are numerous chip architectures, operating systems, and communications protocols; many IoT devices tend to be built around Arm®-based architectures, and if they run an OS it tends to be a Linux distribution.

This commonality is good in many ways; lower costs and faster development times; it also comes with quite a few negatives. Attack vectors tend to become "one-size-fit-all", especially for devices running a Linux-based OS. To mitigate the threats associated with widespread devices that share a common architecture, developers should implement the following "quick win" security design principles:

- Do NOT hardcode passwords into the firmware. Also, do not use a common default password for all devices. Require the user to create a custom username and password during device initialization.
- Do NOT enable insecure protocols such as HTTP, FTP, or Telnet by default. Data going off the device via wired or wireless protocols must be strongly encrypted. Avoid "homebrewed" encryption solutions.
- Ship the device with the most restrictive configuration possible and let the end-user make a proactive decision to reduce security-related settings.
- All mechanisms used to access the devices should require authentication and authorization controls. Two-factor authentication (2FA) should be used if practical.
- All user-facing inputs should be filtered to avoid injection-type attacks.
- Implement a secure device management interface for the end-user that will allow them to manage their assets, update devices, monitor devices, and securely decommission the devices that have reached their end-of-life (EOL).
- Over-The-Air (OTA) update mechanisms must be validated on-device. The update files must be sent encrypted en route to the device. Lastly, ensure there are anti-rollback features to prevent a device from being reverted to a previous, insecure firmware.
- If third-party software libraries are used in the design of your device, they must be continually monitored to ensure that third-party updates are integrated and that they do not become deprecated. Abandoned software projects can become nasty vulnerabilities for your device. Change the third-party software default passwords before committing them to your project.
- Limit what sensitive data should be stored on the device. Store such information in a secure enclave only.
- Remember that with the IoT that embedded systems are only one part of a larger ecosystem. Ensure that security is built into the cloud, desktop, and mobile applications as well. Ecosystem security is only as strong as the weakest link.
- Consider establishing a bug bounty program to encourage end-users and security researchers to submit flaws in a secure, responsible manner.

Physical access to a device tends to be the game-over situation for devices. That doesn't mean that there aren't things that can be done to make it harder to physically exploit these types of devices. Entire books have been written on making circuit boards and associated enclosures tamper-resistant but for a few "quick wins" consider the following physical design rules-of-thumb to harden your device:

The development tools you need

Top 5 Development Tools

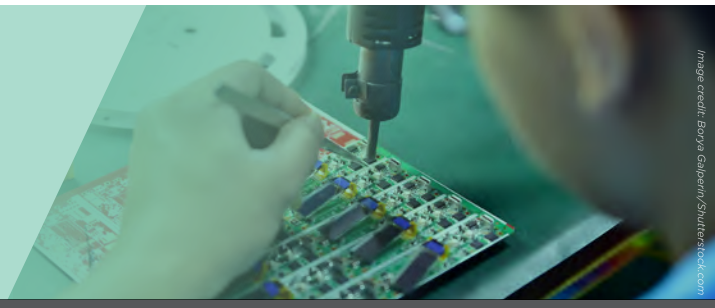


Image credit: Borya Galperin/Shutterstock.com

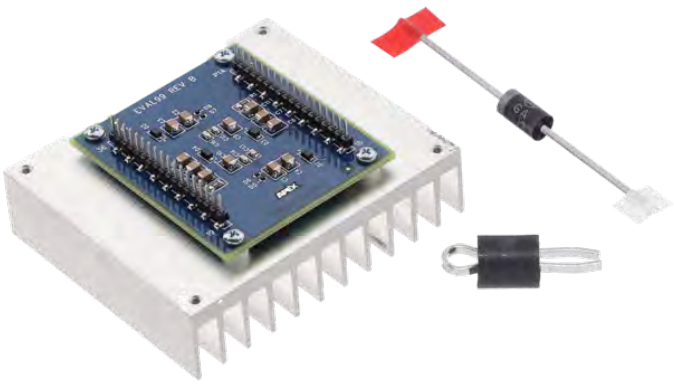
Mouser offers one of the widest ranges of development kits immediately available off-the-shelf to help designers get started. Here, Mouser's Director, Technical Content, EMEA, Mark Patrick, presents his 'Top 5 Pick' of recently-released dev kits. **www.mouser.com/Development-Tools-Center**

Need a power amp?

Apex Microtechnology EK87 Evaluation Kit

This eval kit includes everything needed for rapid prototyping with the PA166 multi-purpose power amplifier IC. It offers flexibility in connecting inputs, measuring outputs, and conditioning signals to the specific application environment. The EK87 comes with an additional prototyping area that allows analyzing many standard or proprietary circuit configurations. Onboard temperature sensing terminals enable real-time output stage junction temperature monitoring. The kit provides flexibility for modifying the gain in an optional improved Howland current pump configuration feature as well as in inverting or non-inverting mode. Typical applications include high-density voltage or current sources, electrostatic transducers, electrostatic/electromagnetic deflection, deformable mirror focusing, and piezoelectric positioning.

FIND OUT MORE >>



3-phase inverter board

STMicroelectronics EVLDRIVE101-HPD reference

This reference design board is an extremely compact, three-phase inverter for brushless motors based on the STDRIVE101 combined with the STM32G071KB microcontroller. This board offers versatile compatibility with a wide range of driving techniques, with FOC and trapezoidal control, as well as sensored and sensorless. The EVLDRIVE101-HPD reference design board provides a ready-to-use and flexible solution to address the needs of high-output current demands in battery-powered three-phase applications. This board integrates a rapid power-on circuit, facilitating seamless battery connection/disconnection, thereby extending the duration significantly. Typical applications include battery-powered home appliances, power tools, fans, industrial automation, drones, and robotics.

FIND OUT MORE >>

Haptic starter kit

The EPCOS / TDK PowerHap starter kit provides users with the first impression of the haptic feedback with PowerHap piezo actuators.

This kit demonstrates how mechanical integration works and offers a reference design that can be adapted to various applications. The PowerHap starter kit consists of a seamless button assembly, a round button assembly, the BOS1901-Kit driver board from Boréas Technologies, a USB cable, a quick-start user guide, and additional PowerHap devices, including an FPC connection cable.

Now at Mouser, the PowerHap kit is equipped with PowerHap actuators that are suitable for various applications such as automotive, smartphones, displays, tablets, household appliances, ATMs, vending machines, game controllers, VR gloves, industrial equipment, and medical devices.

Click for More Information

Rapid development of high precision GNSS systems

The XPLR-HPG-2 Explorer kit from u-blox provides a compact development and prototyping platform for cm-level accuracy positioning applications, such as autonomous robotics, asset tracking and connected health.

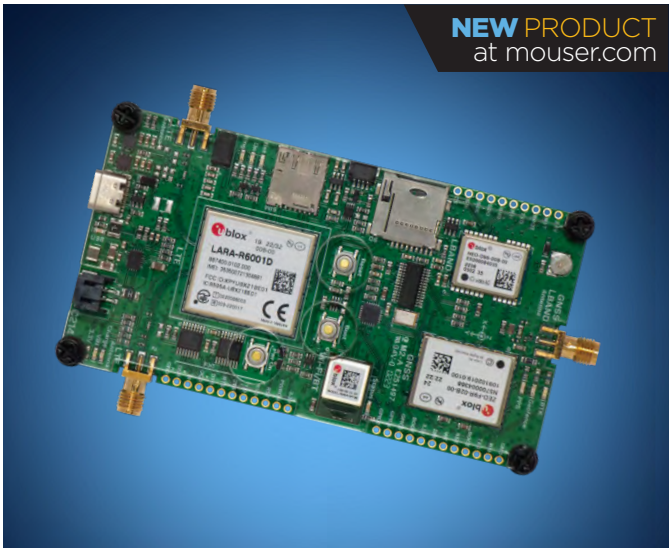
Avaialble from Mouser, the Explorer kit is populated with four modules:

The ZED-F9R is a high-precision sensor fusion module with 3D sensors and a multi-band GNSS receiver. It provides a reliable multi-band RTK turnkey solution with up to 30 Hz real-time position update rate and complete GNSS carrier raw data.

The NEO-D9S module is a satellite data receiver for the L-band correction broadcast, enabling global access to centimeter-level GNSS corrections. The module implements u-blox security principles and advanced security features, including signature, anti-jamming, and anti-spoofing mechanisms, thus allowing reliable GNSS positioning in end-user products.

The LARA-R6001D is a compact LTE Cat 1 multi-mode module offering global coverage. The module is also secure by design with secure boot and secure updates and is certified according to RED cybersecurity requirements.

The NINA-W106 is a small industrial module that integrates a powerful dual-core 32-bit microcontroller with 802.11b/g/n Wi-Fi® and dual-mode Bluetooth® v4.2.



Equipped with its GNSS and communication modules, the XPLR-HPG-2 Explorer kit can access correction data from a satellite broadcast via L-band satellite GNSS receiver or IP connectivity using LTE or Wi-Fi. The NINA-W10 wireless MCU module with Bluetooth and Wi-Fi connectivity runs the HPG software and controls the communications between the u-blox wireless modules. PointPerfect, the u-blox GNSS augmentation service, provides correction data delivered via the Thingstream IoT service delivery platform. The XPLR-HPG-2 also supports the Networked Transport of RTCM via Internet Protocol (NTRIP), so it can also be used with other error correction services.

Click for More Information