April,2023

ICT 6641 Advanced Embedded System Design

Lecture#1: Introduction

S. M. Lutful Kabir, *PhD*
Professor, BUET

ICT 6641: Syllabus

- Hardware design for embedded systems;
- Software development for embedded systems;
- Sensors and Transducers for embedded systems;
- Case study on advanced embedded system;
- Introduction to digital control;
- Its use within embedded systems;
- Case study on digital control in embedded systems;
- Design examples.
- Network based embedded systems;

Introduction

- Microprocessor has revolutionized various industries and our day-to-day life
- Its use in control, monitoring, measurement and signal processing has made a breakthrough in electronic industry
- Now-a-days you won't be able to name one electronic device in which microprocessor or its derivative has not been used
- This course is designed to make you familiar with this remarkable piece of wonder

3

Historical Backgroud

- 1949: Invention of Transistor
- 1959 **:** Invention of Integrated Circuit
- 1971: Invention of uP
- 1976: Invention of first uC

4

Invention of Transistor

- In 1947, Willium Schockley and his colleagues in Bell laboratory invented transistor
- That introduced a new era in electronic industry
- Transistor is being used in all electronic circuits replacing its rival vacuum tube
- Vacuum tubes are bulky, consumes lots of power, unreliable, occupies lots of space etc.
- So, a lot of research and development efforts were put in the area of solid state electronics
- The basic idea was that semiconductor materials like germenium, silicon can carry currents and the current can be controlled by the injuction of some impurity

5

Invention of Integrated Circuit

- Very soon in 1959, another invention made revolution in electronic industry
- Some scientists in Fairchild Semiconductor invented planner technology of transistors which ultimately led to the invention of Integrated Circuit Technology
- An integrated circuit is one in which a number of transistors or components can be fabricated in a single silicon wafer
- The rate of production of IC was progressed rapidly over next few years

6

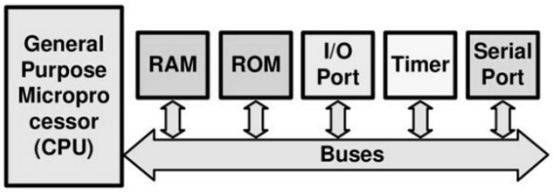
Evolution of IC Technology

Year	Technology	Number of Devices	Typical Products
1947	Invention of Transistor	1	Transistor
1950-1960	Discrete Components	1	Junction Diode and Transistor
1961-1965	SSI	10-100	Planner Devices, Logic Gates, FFs
1966-1970	MSI	100-1000	Counter, MUX, Decoders
1971-1979	LSI	1000-20000	8-bit uP, RAM, ROM, DSP, RISC
1980-1984	VLSI	20000-50000	16-bit, 32-bit uP
1985-	ULSI	Greater than 50000	64 bit uP

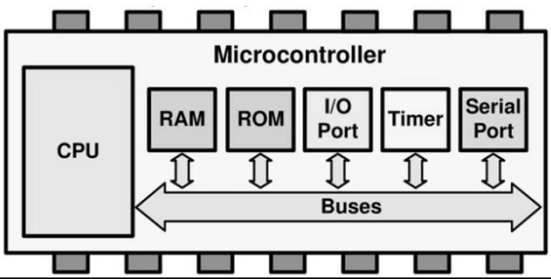
7

Microprocessor vs Microcontroller

- General Purpose Microprocessor

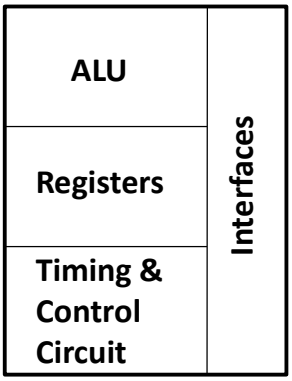


- Microcontroller



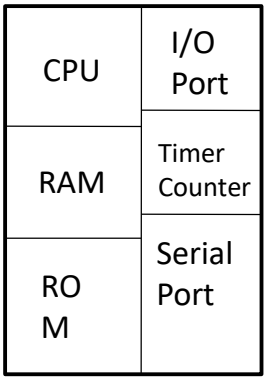
Evolution of Microprocessor

- In 1971, first microprocessor was developed by Intel, this particular device is 4 bit uP called 4004
- By microprocessor we mean that all the components one finds in a CPU (ALU, Register, Timing & Control Unit and the Interfacing circuits) are put together in a single chip
- That is a uP can be considered as “CPU in a Chip”
- Other companies like Texus Instruments, Fairchild etc have started manufacturing uP
- The reason was that although 4004 was developed for making calculator but they found its potential for making any intiligent electronic system.



Evolution of Microcontroller

- Very soon in 1976 first microcontroller was produced again by Intel, 8048
- Microcontroller is somewhat different from microprocessor in a sense that not only the CPU but also RAM, ROM, I/O Ports, Timer & Counter, Serial Port all put together in a single chip
- So, it is called as “Computer on a chip”

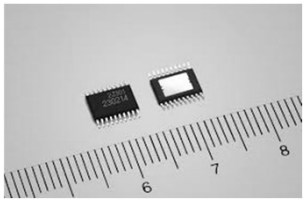


Why uP and uC are so popular?

- Small Size
- Lower Cost
- Higher Reliability
- Lower Power Consumption
- Higher Versatility
- More Powerful

Smaller Size

- A tiny chip can contain hundreds of thousands of transistors and components
- The size of present day microprocessor has now reduced to less than an inch
- With the advent of VLSI technology the microprocessors/ microcontrollers now come with a large numbers of pins



Lower Cost

- With the advent of complex technology, microprocessor microcontroller's cost does not increase proportionally
- Rather its cost per function has decreased gradually
- As the material cost is negligible in comparison with the initial masking cost, per unit cost decreases as the volume of production increases

Higher Reliability

- With the advent of more sophisticated fabrication technology and testing system microprocessor/ microcontroller manufacturing process is now very reliable
- Moreover, the microprocessor/ microcontroller based system now-a-days requires less number of chips and so wiring requirement is less making the overall system more reliable

Power consumption

- Present day microprocessors/ microcontrollers consume very low power in the order of milliwatt
- This is because of the fact that they are manufactured by transistors using CMOS technology which requires very low power

Versatility

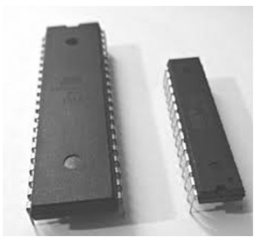
- Since the program stored in a microprocessor/ microcontroller system is rewritable hence one system can be reused by loading a new program
- This is especially useful when a new system is being developed. Frequent change in programming is necessary to fix bugs if any

More powerful

- Present day microprocessor/ microcontroller is more powerful than the previous one, w.r.t processing speed, memory and I/O pins
- This becomes possible because of increase in data addressible bit size
- Now 32 bit processor is common in the market
- A 32 bit microprocessor/ microcontrolleris powerful than the then main frame computer

17

Microcontroller and some of their applications



Typical Applications of uP

- Microprocessor is basically a CPU, using CPU general purpose computer can be built
- Desktop computers, PCs, Laptops, Workstations, Servers, Supercomputers
- We are in the era of 32- and 64-bit microprocessor, so we can produce very powerful computers

19

Typical Application of uC

- Using microcontrollers embedded systems are produced
- An embedded system is a combination of hardware and software built for a specific application
  - Consumer electronics making toys, cameras, camcorders, robots
  - Consumer products - washing machines, microwave ovens etc.
  - Instrumentation - oscilloscopes, various medical equipments,
  - Process control - data aquisition, controlling various industries
  - Communication – Telephone, answering machines
  - Office equipments – Fax, printer, small PABX
  - Micromedia application – Cell phones, PDAs, teleconferencing equipment.

Popular Microcontroller Manufacturers

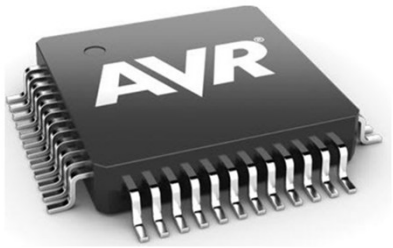
8051

8051 Microcontroller

PIC

PIC Microcontroller

AVR



AVR Microocntroller

ARM

ARM Microcontroller

	8051	PIC	AVR	ARM
Bus width	8-bit for standard core	8/16/32-bit	8/32-bit	32-bit mostly also available in 64-bit
Communication Protocols	UART, USART,SPI,I2C	PIC, UART, USART, LIN, CAN, Ethernet, SPI, I2S	UART, USART, SPI, I2C, (special purpose AVR support CAN, USB, Ethernet)	UART, USART, LIN, I2C, SPI, CAN, USB, Ethernet, I2S, DSP, SAI (serial audio interface), IrDA
Speed	12 Clock/instruction cycle	4 Clock/instruction cycle	1 clock/ instruction cycle	1 clock/ instruction cycle
Memory	ROM, SRAM, FLASH	SRAM, FLASH	Flash, SRAM, EEPROM	Flash, SDRAM, EEPROM
ISA	CLSC	Some feature of RISC	RISC	RISC
Memory Architecture	Harvard architecture	Von Neumann architecture	Modified	Modified Harvard architecture
Power Consumption	Average	Low	Low	Low
Families	8051 variants	PIC16,PIC17, PIC18, PIC24, PIC32	Tiny, Atmega, Xmega, special purpose AVR	ARMv4,5,6,7 and series
Community	Vast	Very Good	Very Good	Vast
Manufacturer	NXP, Atmel, Silicon Labs, Dallas, Cyprus, Infineon, etc.	Microchip Average	Atmel	Apple, Nvidia, Qualcomm, Samsung Electronics, and TI etc.
Cost (as compared to features provide)	Very Low	Average	Average	Low
Other Feature	Known for its Standard	Cheap	Cheap, effective	High speed operation Vast
Popular Microcontrollers	AT89C51, P89v51, etc.	PIC18fXX8, PIC16f88X, PIC32MXX	Atmega8, 16, 32, Arduino Community	LPC2148, ARM Cortex-M0 to ARM Cortex-M7, etc.

Center of the Development: STM32

- A family of 32-bit microcontrollers
- Manufactured by STMicroelectronics.
- Based on the ARM Cortex-M architecture
- Widely used in
  - embedded systems,
  - IoT devices, and
  - other applications requiring high-performance and low power consumption.
- Offers a wide range of products, with
  - various options for memory,
  - performance, and
  - peripherals.

Center of the Development: STM32 (continued)

- Available in various package sizes, from
  - small and low-power devices to
  - high-performance devices
  - with multiple cores and
  - advanced peripherals.
- Designed to be easy to use and integrate into a variety of systems.
- To help developers get started quickly and easily. STMicroelectronics provides
  - a comprehensive ecosystem of development tools,
  - software libraries, and
  - reference designs.

Key features of the STM32

- Low power consumption - ideal for battery-powered devices and other low-power applications.
- High performance - the ARM Cortex-M architecture, enabling the microcontrollers to handle complex tasks and real-time applications.
- Flexible peripherals - the microcontrollers have a wide range of peripherals, including GPIO, timers, ADCs, DACs, and communication interfaces such as UART, SPI, I2C, USB, and Ethernet.
- Rich software ecosystem - the microcontrollers are supported by a range of development tools and software libraries.

Integrated Development Environment (IDE)

- There are some different options for IDEs. Here are a few of them:
  - Eclipse
  - ARM Embed
  - Keil
  - IAR Systems
  - Atollic TrueStudio
  - STM32CUBE IDE
- **STM32CUBE IDE** is the software tool we'll be using.
- The toolchain provides features to ease and accelerate the development, debugging, and testing tasks.

key features of STM32CubeIDE

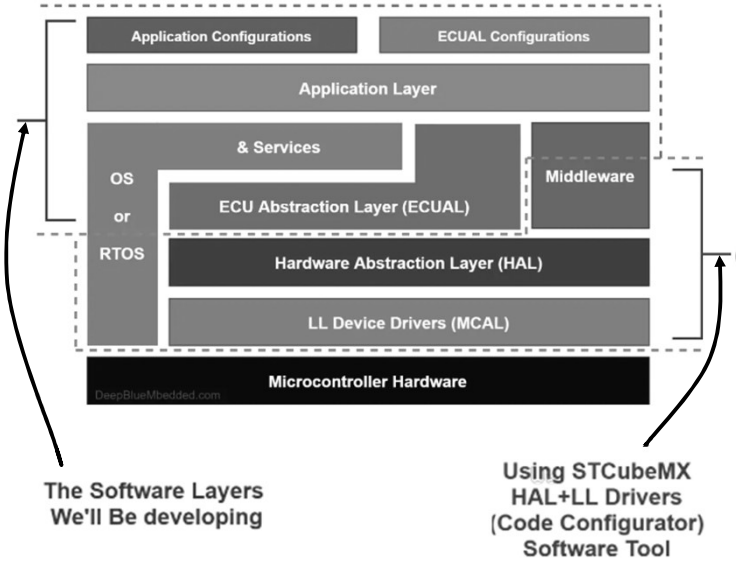
- Code generation: STM32CubeIDE includes a code generator that can help you quickly generate code for your STM32 microcontroller. Debugger: STM32CubeIDE includes a powerful debugger that will help you identify and fix bugs more quickly and efficiently.
- Project management: STM32CubeIDE allows you to manage your projects easily. You can create, edit, and organize your projects in a user-friendly interface.
- Support for multiple languages: STM32CubeIDE supports multiple programming languages, including C, C++, and Assembly.
- Library management: STM32CubeIDE comes with a library manager that allows you to easily manage your libraries and dependencies.

Downloading the software for the IDE

- https://www.st.com/en/development-tools/stm32cubeide.html
- From the page select the correct software for your operating system. For windows choose the bottom most one.

Firmware Development Level of Abstraction

- The software layered architecture diagram is given.
- We won't be developing LL drivers at the register level.
- However, we'll be using the LL+HAL device drivers provided by STMicroelectronics.
- So we can dedicate the development effort to the application layer and middleware, mostly the ECU abstraction layer (ECUAL) drivers.



Advantages of Abstraction

- Advantage #1:
  - You'll learn how to develop reusable configurable firmware drivers for different modules, sensors, and interfaces.
  - All of which will be dependent on the STM's HAL+LL drivers that have uniformed APIs across the entire portfolio of STM's STM32 families.
- Advantage #2:
  - Will have an embedded software stack that can potentially run on any STM32 microcontroller with very little effort.
  - This is really helpful if you're designing your own STM32-based PCB boards and projects with any microcontroller part,
  - Having portable reusable firmware drivers is key in shortening the development time.

STMCube MX Software

- https://www.st.com/en/development-tools/stm32cubemx.html
- Like the previous case, choose the software for your operating system.
- This is the second software tool you need to download and install.
- Obviously, we'll use the CubeMX GUI app to setup and configure the low-level hardware and peripherals.
- It also helps you configures the clock tree of the microcontroller to decide on the various clock speed for the system, buses, and peripherals.
- At the end of the configuration process, it generates the project folder in the specified directory.
- Then you click a button in order to launch it in the Cube IDE and start developing your project right away.

ST-link V2 Driver

- You'll finally need to make sure that the driver for the ST-Link v2 programmer/debugger is correctly installed and it's assigned a virtual COM port by your operating system.
- For any Nucleo or discovery board you'll not be in need to do this step.
- However, for the blue pill, you'll need an external USB ST-Link v2 debugger and it may not install the driver automatically once connected to your USB port.
- So here is the link:
- https://www.st.com/en/development-tools/stsw-link009.html

Thanks