BITS Pilani presentation

BITS Pilani
Pilani Campus

Dr Ramani Kalpathi
Automotive Electronics

innovate achieve lead

BITS Pilani

Pilani Campus

AELZG512, Embedded System Design
Lecture No. 1

Introduction to Embedded Systems

- Characteristics and Embodiments of Embedded System
- Classification of Embedded Systems
- Introduction to Hardware and Software components

Hardware Components of Embedded System

- Introduction to Processor Architectures
- Memory Types Organization, Cache
- Interrupts
- Basic peripherals like Timers , ADC/DAC

Software components of Embedded System

- RTOS & Tasks
- Introduction to SOC design, Embedded System Design

What are Embedded Systems?

Embedded Systems are products designed with computational intelligence and interfaces to the physical environment to perform specific functions. These products are designed to meet a set of specified requirements derived to provide solutions to real world problems

The computational intelligence in embedded systems refers to the use of microprocessors, microcontrollers or programmable FPGA devices that form the heart of embedded systems.

Microcontrollers in embedded systems

Microcontrollers are interconnected along with discrete devices and sensors to form the hardware part of the embedded system. The intelligence for the embedded system is the software program that is loaded into the memory of the microprocessor peripherals. A hand held cell phone is a typical example of an embedded system that has a central microcontroller and peripheral devices such as a keypad, display and audio speakers.

Characteristics

Specific Function: A camera has the primary function of taking pictures
Constraints: Size, cost , performance
Real Time System: Interact with environment in real time to adjust performance
Reliability: Repeatable fail-safe performance
Complex Algorithms: Computational and logic capability

Simple Embedded Systems

Digital clock, calculator, digital thermometer

One microcontroller, display, few buttons, switches

Simple program with a few computations

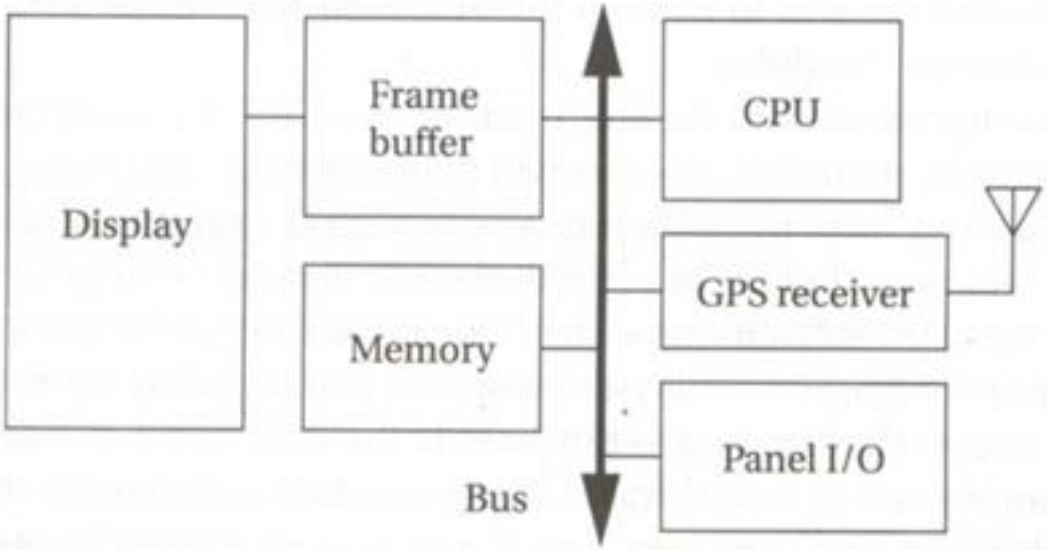
Complex Embedded Systems

Autonomous vehicle systems

Multiple Processors, Various communication interfaces, Real Time Operating Systems

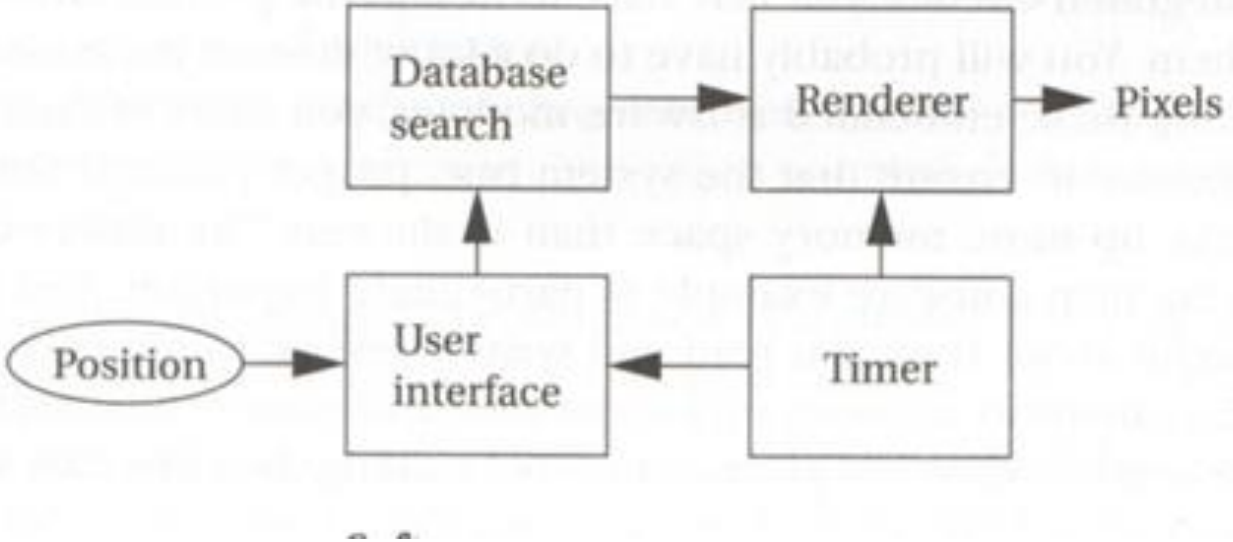
Hardware and Software Components (Navigational embedded System)

Hardware: CPU with memory and I/O devices, different types of memory – frame buffer for map image and memory for CPU program, data.

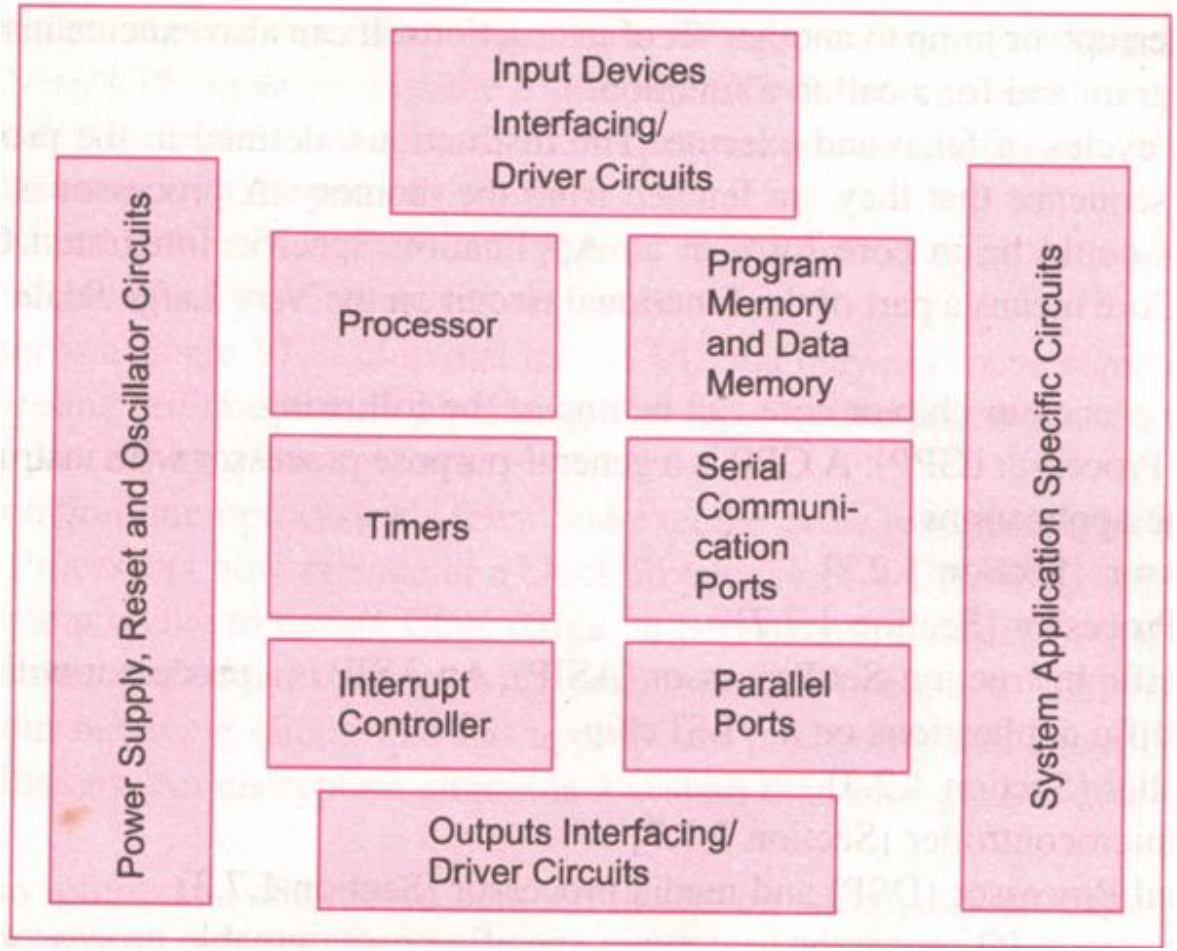


Navigational System Software Components

Software: User interface, search for database to get geographical information, rendering algorithms to display the data, position retrieval from GPS device, timer functions to perform the operations synchronized to time.

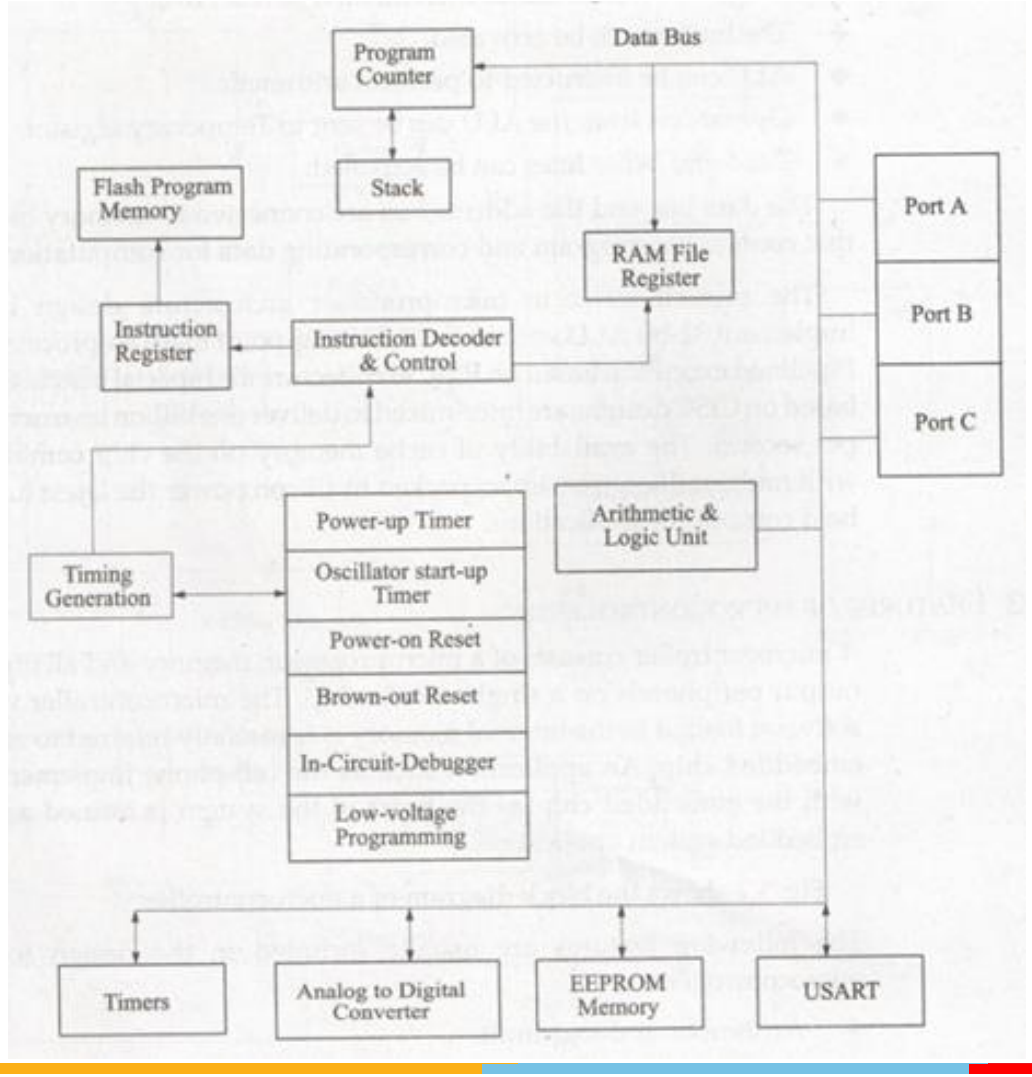


Embedded System Hardware Components



*Rajkamal – Embedded Systems

Block Diagram of a Microcontroller



Types of Memory

OTPROM – One Time Programmable Read Only memory

EPROM – Electrically Programmable Read Only Memory

EEPROM – Electrically Erasable Programmable…

RAM (SRAM,DRAM) – Random Access Memory

Flash memory – Writeable using fast flash operations

Memory - Usage

OTPROM- Cheapest and used for high volume applications to store programs

EPROM – Needs an UV eraser to erase programs (not common now)

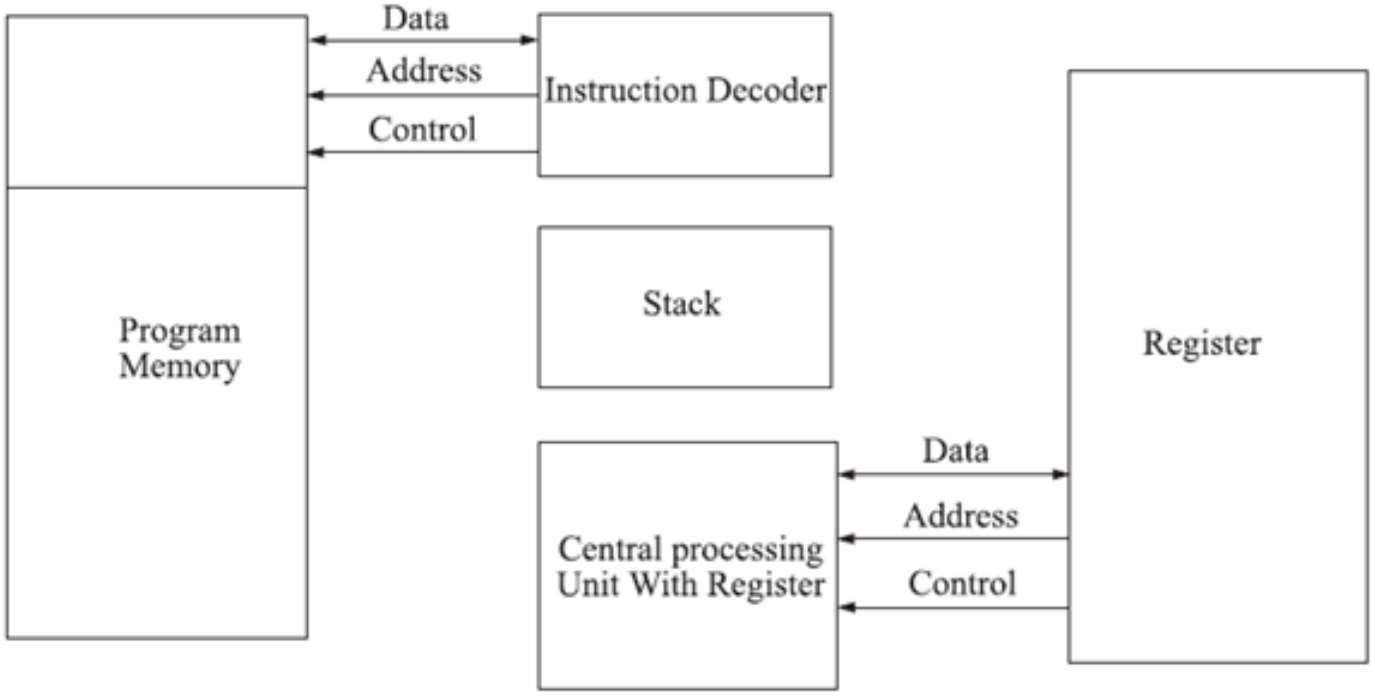
EEPROM – Data memory that can be retained even if the power is turned OFF

Flash memory – Programs are stored in this area

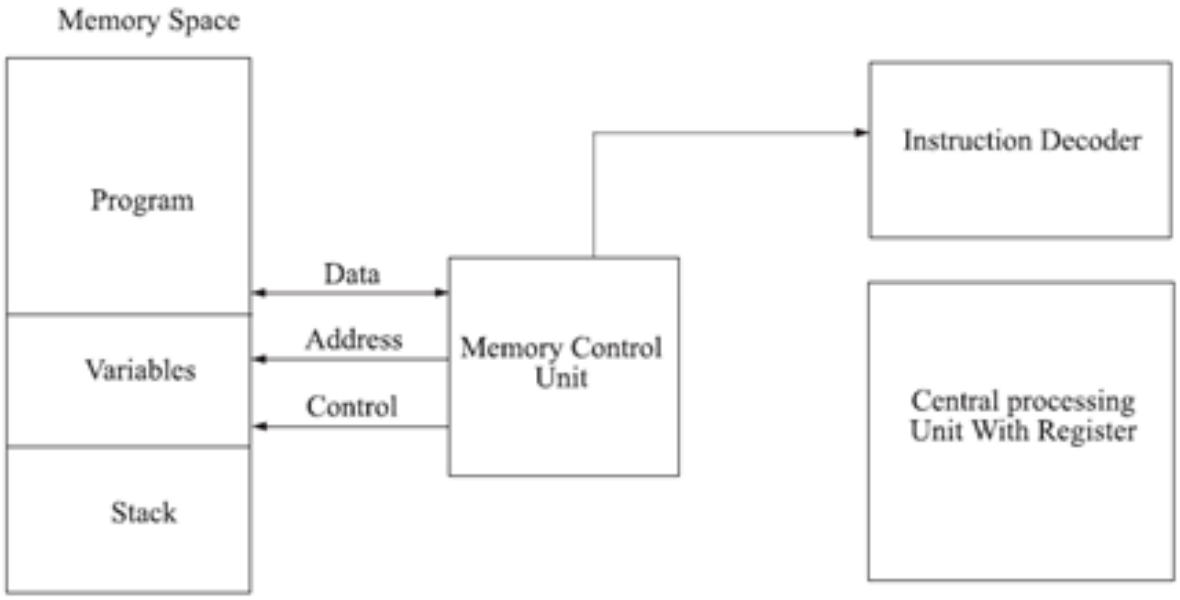
RAM – Variables declared in C programs or assembly that get modified as the program executes are stored in this memory area

Cache memory – Located close to the CPU and has fast access times for read and write operations

CPU Architecture-Harvard



Von Neumann Architecture



CISC Architecture- Complex Instruction Set

The features of a CISC computer are listed as follows:

- Large number of instructions.
- Execution time for each instruction may be different.
- Execution time for an instruction may take several clock cycles.
- Single instruction will perform a complex operation.
- Efficient use of memory space.
- Robustness of instruction set given priority over speed of execution.
- Works faster for complex instructions but may be slow for simple tasks.
- Emphasis on complex hardware to store a large instruction set.
- Number of instructions per program is minimized.

The most common examples of CISC architecture are

- Intel 8080,80286,80386,Pentium
- Motorola 68000
- Zilog Z80

RISC architecture

Reduced Instruction Set Computer (RISC) refers to a type of microprocessor architecture that utilizes a small but highly optimized set of instructions. The design features of RISC processors are as follows:

- One cycle execution time: Most instructions are executed in one clock cycle. This design optimization of each instruction follows a technique called pipelining.
- Pipelining: Different parts of an instruction are executed simultaneously.
- Reduced complexity in hardware.
- Less hardware space for transistors in the silicon for each instruction.
- Large Register set: The architecture uses a large number of registers to handle data instead of relying on the memory storage.
- Variables and intermediate results stored in registers.
- Complex operations are performed as a series of simple RISC instructions.
- Instructions are of fixed length and format.
- Emphasis on software for each task.
- Program consists of large code size.
- Only Load and Store instructions refer to the memory and may take more than one clock cycle.

Atmel 89S51 – Special Function Registers

Symbol	Name	Address (Hex)	Remarks
ACC	Accumulator	0E0	Bit addressable
B	B Register	0F0	Bit addressable
PSW	Program Status Word	0D0	Bit addressable
SP	Stack Pointer	81	
DPTR	Data Pointer 2 Bytes		
DPL	Low Byte	82	
DPH	High Byte	83	
P0	Port 0	80	Bit addressable
P1	Port 1	90	Bit addressable
P2	Port 2	0A0	Bit addressable
P3	Port 3	0B0	Bit addressable
IP	Interrupt Priority Control	0B8	Bit addressable
IE	Interrupt Enable Control	0A8	Bit addressable
TMOD	Timer/Counter Mode Control	89	
TCON	Timer/ Counter Control	88	Bit addressable
T2CON	Timer/ Counter 2 Control	0C8	Bit addressable, 8052 only

Input Output Peripherals

Output Operations: P0,P1,P2 and P3 are four 8-pin peripherals in the 8051 microcontroller. Logic output of 0V or 5V can be output to those pins by using C program or assembly program to write to the special function registers.

Example

P0 = 0x01;

This is a C program statement that will send a hexadecimal value of 0x01 which has a binary equivalent value of 00000001 to the Port output pins of P0.0 to P0.7 resulting in a high logic voltage at P0.0 pin measuring 5V and a low logic voltage measuring 0V at pins P0.1, P0.2,P0.3,P0,4,P0.5,P0.6,P0.7

Output operations

P0 =0x01; // 0000 0001

P0 = 0x02; //0000 0010

P0 = 0x03;// 0000 0011

P0 = 0x08;// 0000 1000

P0 = 0x09 ;//0000 1001

P0 = 0x0A;//0000 1010

P0 = 0x0B;//0000 1011

P0 = 0x0C;//0000 1100

P0 =0x0D; //0000 1101

P0 = 0x0E;//0000 1110

P0 = 0x0F;//0000 1111

P0 = 0x10;//0001 0000

P0= 0x20; //0010 0000……and so on

Programs can be written to make a pin high or low by writing to special function registers. Each microcontroller has output ports that can be configured for such operations

Input Operations

To read the logic voltage applied to a pin from the external circuitry to the microcontroller, the port has to be read by an instruction and its value recorded in a variable.

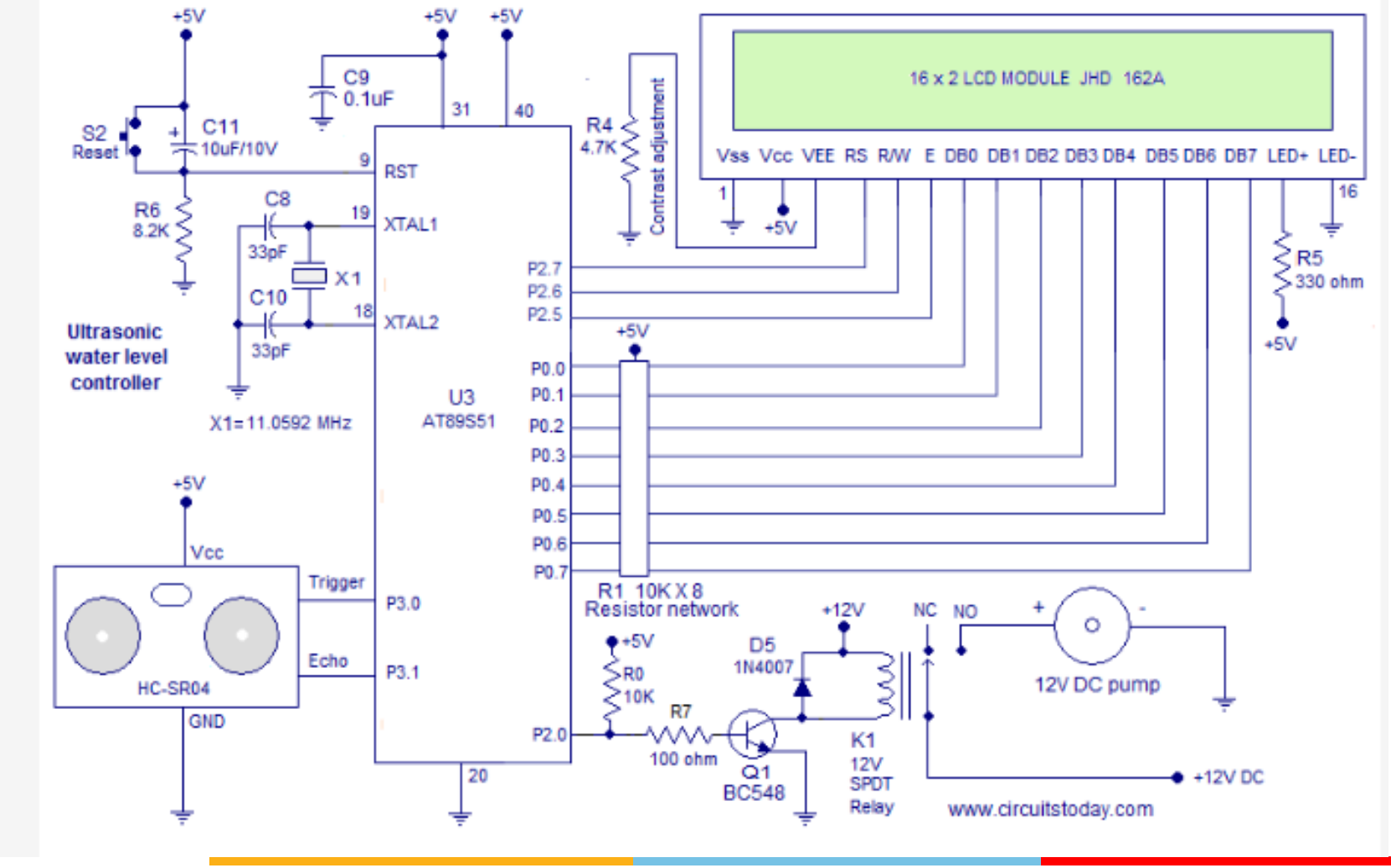
portvalue1 = P1;

In this instruction P1 is a special function register and its pins may be connected to a switch input that may provide 0V or 5V to the individual pins. For example if P1.0 is connected to 5V and P1.7 is connected to 5V and all other pins connected to 0V, then the variable portvalue1 = 1000 0001 in binary

or portvalue1 = 0x81 in hexadecimal number.

Hexadecimal notation is commonly used to write to Special function registers since the code appears compact.

Circuit Diagram Using 8051



Circuit Notes

Pins 31, 40 are connected to 5V power supply

Pin 20 is connected to GND

Pins 18 and 19 are connected to a crystal clock

Pin 9 is connected to a RC circuit with a reset switch

The microcontroller needs these basic connections to start executing a program. Program needs to be loaded into the flash memory of the microcontroller and it will start executing the instructions starting from address 0x0000 sequentially.

The Program counter is loaded with the instruction address 0x0000 upon power up and the microcontroller starts executing instructions from the first address sequentially

Assembly Language Program

innovate achieve lead

```
;Turn on relay connected to Port 0.0 ON and OFF with a short delay
.org 0h    ;signifies start of assembly language routine
start:     mov P0,#00h
           acall delay
           mov P0,#01h ; Turn ON Relay1 connected to P0.0
           acall delay
           sjmp start

delay:     mov r0,#0FFh; Load the hex number FF to register r0
wait:      djnz  r0,wait
           ret
.end  ; signifies end of assembly language routine
```

Interrupts

Interrupts are branches in code execution triggered by hardware events or software events. There are two external event interrupt sources featured by pins INTO and INT1 in 8051 controllers. Once configured in software, a high to low transition in one of these pins will cause the program execution to branch to a specific location called the interrupt vector address. The program counter will point to this interrupt vector address and execute what is called an interrupt subroutine. After execution of the interrupt subroutine, the Program counter points to the address from where it branched off before the interrupt occurred and continues program execution.

Peripherals-GPIO, Timers

Microcontrollers communicate and control the outside world through these peripherals

Input/output ports – Pins used to turn ON, turn OFF, read logic high or low levels – Used for controlling relays, transistors, display devices

Timer/Counter – Peripheral used to count events, find time between events, schedule events, provide periodic time intervals for program scheduling

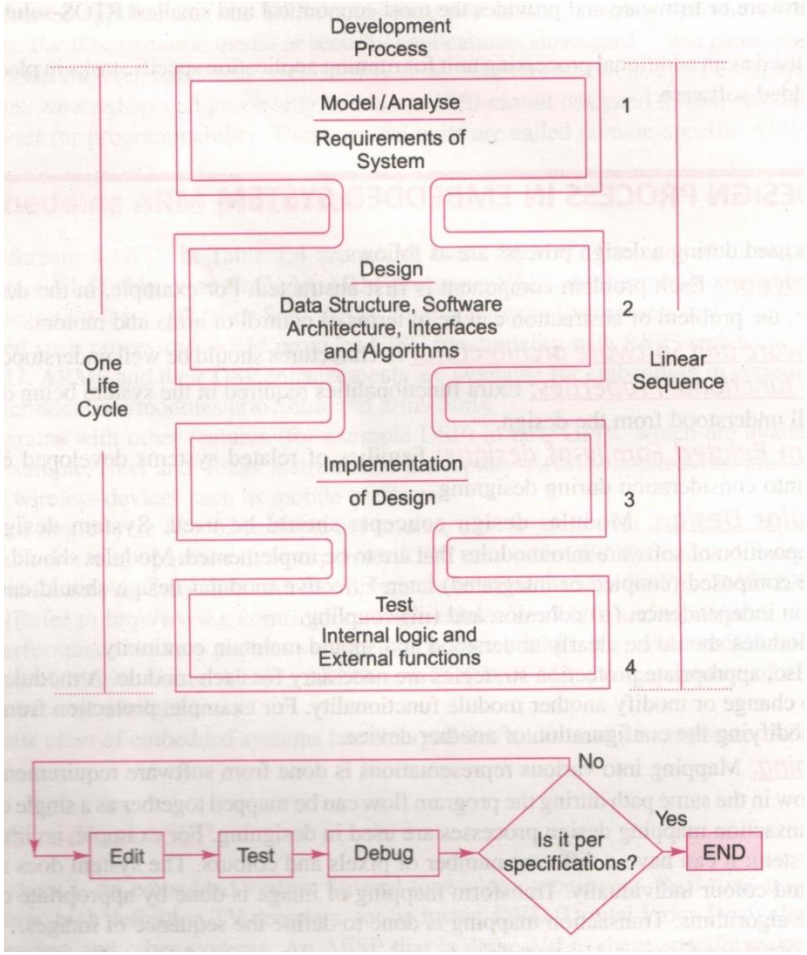
ADC, DAC, Serial

ADC – Analog to Digital Converter is a peripheral used to convert the instantaneous value of a time varying analog signal to a binary number. For example a 8 bit analog to digital converter will convert a 2.5V signal to 1000 0000 (128 in decimal) and a 5V signal input to 1111 1111 (255 in decimal). These peripherals are used in most measurement and control applications.

DAC – Digital to Analog Converter is a peripheral used to convert digital value to analog value. These peripherals are used for waveform generation applications for example.

Serial Ports –Microcontrollers use serial peripherals to transfer data using a minimal number of connections by transmitting data one bit after another.

Software Design Flow



*Rajkamal, Embedded Systems

Real Time Operating Systems

A real-time operating system (RTOS) is an operating system (OS) intended to serve real-time application process data as it comes in, typically without buffering delays. Key factors in an RTOS are minimal interrupt latency and minimal thread switching latency. An RTOS is valued more for how quickly or how predictably it can respond than for the amount of work it can perform in a given period of time.

For embedded devices, the *general* rule is that an RTOS is used when the application needs to do more than a few simple actions. An RTOS allows an application to be structured in a manner that scales as more application/system features are added (e.g. communication stacks, power management, etc.).

RTOS Goals

Small latency: Minimum delay time from one task to another.

Determinism: Time taken for execution of a task or thread needs to be predictable so that deadlines are known.

Structured Software: Software tasks can be organized, added or deleted based on the system requirement.

Scalability: RTOS must be able to scale from a simple application to a complex one with stacks, drivers, file systems.

Offload development: An RTOS manages many aspects of the system which allows a developer to focus on their application. For example an RTOS, along with scheduling, generally handles power management, interrupt table management, memory management, exception handling.

System On Chip

A system on a chip (SoC) is an integrated circuit that integrates all components

A CPU, memory, GPIO and secondary memory are placed on a single substrate or microchip.

Digital logic, analog circuits, mixed signal and radio frequency circuits may be added into the single substrate.

SOC devices consume less power, less area and are much smaller than multi-chip designs.

SoCs are common in the Smartphone applications, Internet of things applications.