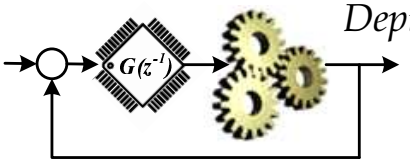
Chapter 2
Microcontroller System Hardware Fundamental

Assoc. Professor Dr. Rosbi bin Mamat

rosbi@fke.utm.my



Dept. of Control & Mechatronics Engineering
Faculty of Electrical Engineering
Universiti Teknologi Malaysia

2.1 Intro to ATMEL AVR μc

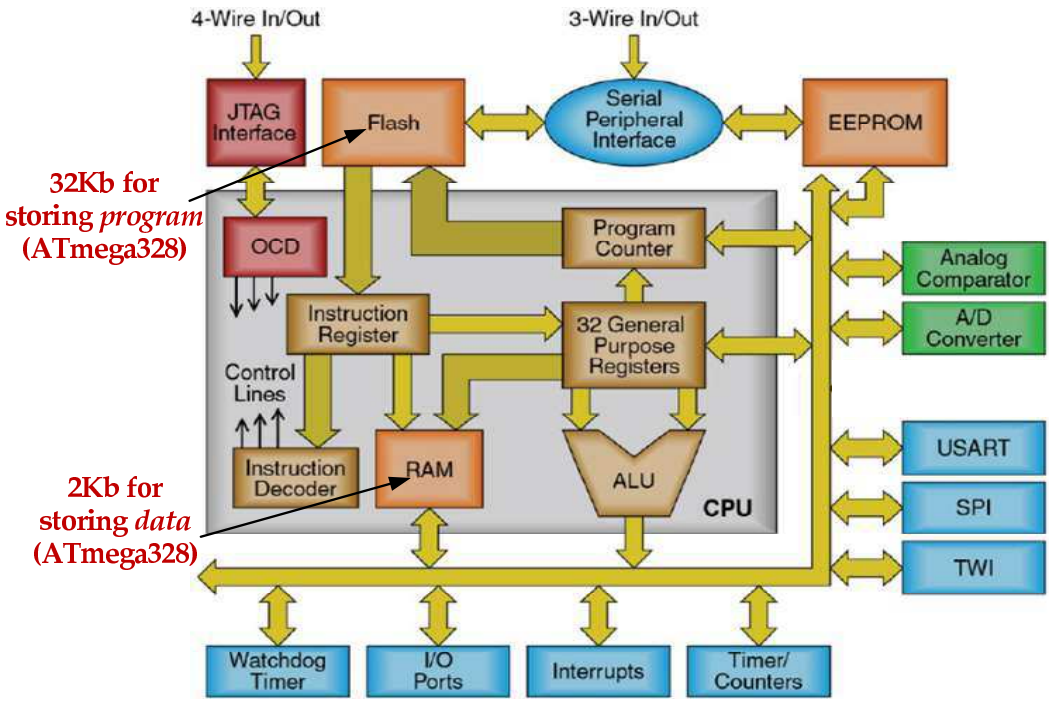
- Features of ATMEL AVR μc:
  - Low-power 8-bit μc with Harvard RISC architec. – 131 instructions mostly execute in single-clock cycle, performance up to 20 MIPS at 20 MHz.
  - Register-based & C language friendly architec. – 32 x 8 general purpose working reg.
  - In-System Self-programmable (ISP) Flash program memory for storing **program code**
  - EEPROM & internal SRAM for storing **data**
  - Internal peripherals: *Timer/Counters, Analog comparator, PWM Channels, 8-channel 10-bit ADC, bidirectional I/O ports, I^2C, SPI, USART communication interfaces, Calibrated RC oscillator.*

2.2 ATMEL ATmega328 μc

- Features ATmega328:
  - 32K Bytes of ISP Flash program memory (Write/Erase Cycles: 10,000)
  - 2K Bytes internal SRAM
  - 1024 Bytes EEPROM (Write/Erase Cycles: 100,000)
  - JTAG programming/debugging interface
  - 6-channel, 10-bit Analog-to-Digital converter (ADC)
  - 2× 8-bit Timer/Counters & 1× 16-bit Timer/Counter
  - 6 PWM channels
  - Serial Comm. interfaces: USART, SPI, I^2C
  - Watchdog Timer, Analog Comparator, Reset Brown-out detection, Internal RC oscillator, Power Save modes
  - Speed Grades: 0 – 20MHz (20MIPS)

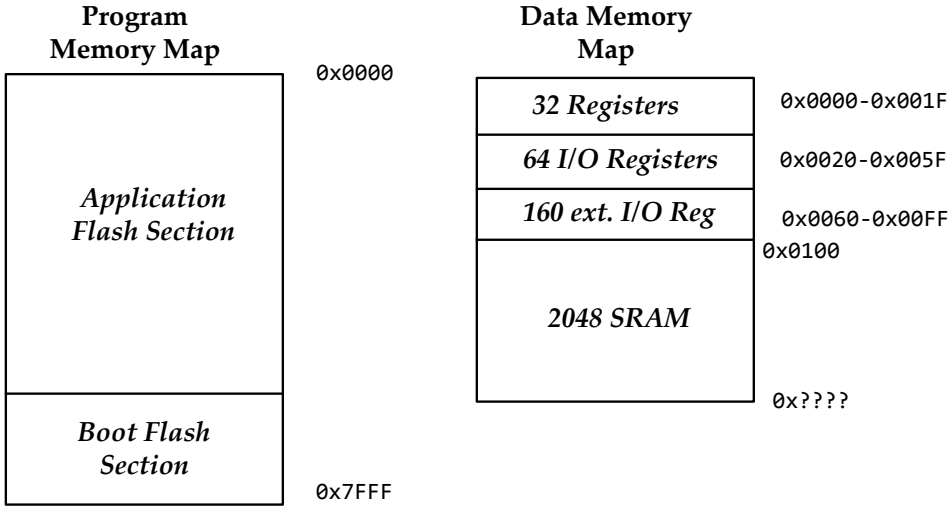
2.2 ATMEL ATmega328 μc

- Internal architecture:



2.2 ATMEL ATmega328 μc

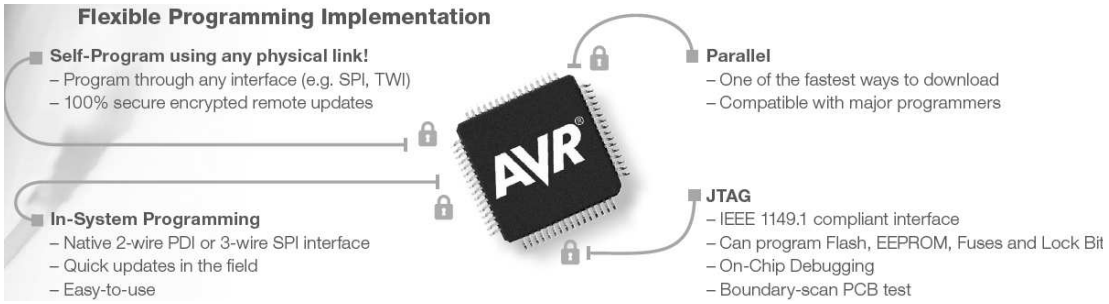
- Memories map in ATmega328:



- *Exercise: If the boot flash start address is 0x7C00, what is the size of boot flash section ? (in Kb)*
- *Exercise: Calculate the end address of SRAM.*

2.2 ATMEL ATmega328 μc

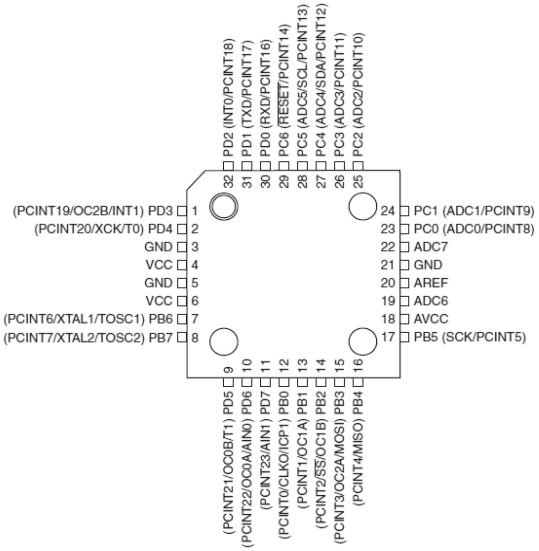
- How to *program* memories (Flash & EEPROM) in ATmega328 ?



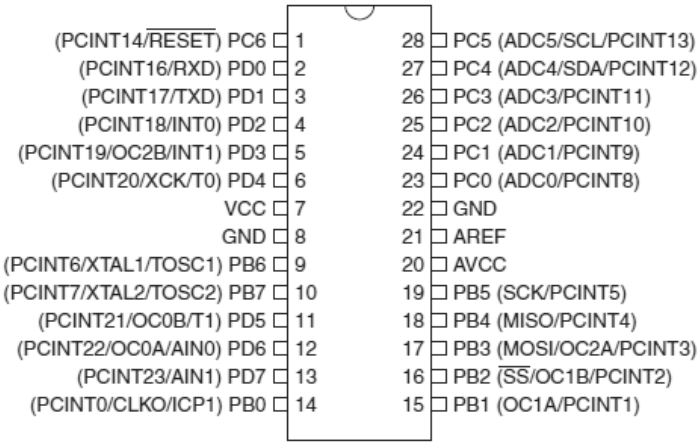
- Parallel method, JTAG interface & ISP – Require programmer hardware that support them.
- Self-Program using any physical link – Program with the help of resident *bootloader.* The bootloader must be preprogrammed in the Boot Flash Section

2.3 ATmega328 Pinout & Descriptions

TQFP/MLF

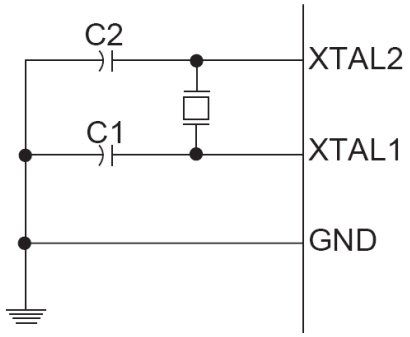


PDIP



2.3 ATmega328 Pinout & Descriptions

- `VCC` & `GND` – Digital supply voltage for ATmega328. Typical operating voltages are 2.7 – 5.5V. Should never exceed 6.0V !
- `XTAL1` & `XTAL2` – Connections to ceramic/crystal on-chip oscillator for generating internal clocks. Crystal connection as below where typical values for `C1`, `C2` are 22pF. Max crystal freq. is 20MHz.



2.3 ATmega328 Pinout & Descriptions

- $\overline{\texttt{RESET}}$ – A low level on this pin for longer than 1.5μSec length will generate a reset. During Reset, all I/O Reg. are set to their initial values, & the program starts execution from the **Reset Vector** (`0x0000`) or Boot Flash start addr. (`0x7000`, `0x7800`, `0x7C00`, `0x7E00`) if **Boot Reset Fuse** is programmed.
- `AVCC` – the supply voltage pin for the ADC. Should be externally connected to V_{CC}.
- `AREF` – the analog reference pin for the ADC. Connect to suitable voltage if external ref is chosen.

2.3 ATmega328 Pinout & Descriptions

- Port B (`PB0-PB7`) – serves as an 8-bit **bi-directional** digital I/O port with optional internal pull-ups. Port B pins are tri-stated when reset. Port B can be programmed to serve as **alternate function**:

Port Pin	Alternate Functions
PB7	XTAL2 (Chip Clock Oscillator pin 2) TOSC2 (Timer Oscillator pin 2) PCINT7 (Pin Change Interrupt 7)
PB6	XTAL1 (Chip Clock Oscillator pin 1 or External clock input) TOSC1 (Timer Oscillator pin 1) PCINT6 (Pin Change Interrupt 6)
PB5	SCK (SPI Bus Master clock Input) PCINT5 (Pin Change Interrupt 5)
PB4	MISO (SPI Bus Master Input/Slave Output) PCINT4 (Pin Change Interrupt 4)
PB3	MOSI (SPI Bus Master Output/Slave Input) OC2A (Timer/Counter2 Output Compare Match A Output) PCINT3 (Pin Change Interrupt 3)
PB2	$\overline{\text{SS}}$ (SPI Bus Master Slave select) OC1B (Timer/Counter1 Output Compare Match B Output) PCINT2 (Pin Change Interrupt 2)
PB1	OC1A (Timer/Counter1 Output Compare Match A Output) PCINT1 (Pin Change Interrupt 1)
PB0	ICP1 (Timer/Counter1 Input Capture Input) CLKO (Divided System Clock Output) PCINT0 (Pin Change Interrupt 0)

2.3 ATmega328 Pinout & Descriptions

- Port C (PC0-PC6) – serves as an 8-bit **bi-directional** digital I/O port with optional internal pull-ups. Port C pins are tri-stated when reset. Port C can be programmed to serve as **alternate function**:

Port Pin	Alternate Function
PC6	$\overline{\text{RESET}}$ (Reset pin) PCINT14 (Pin Change Interrupt 14)
PC5	ADC5 (ADC Input Channel 5) SCL (2-wire Serial Bus Clock Line) PCINT13 (Pin Change Interrupt 13)
PC4	ADC4 (ADC Input Channel 4) SDA (2-wire Serial Bus Data Input/Output Line) PCINT12 (Pin Change Interrupt 12)
PC3	ADC3 (ADC Input Channel 3) PCINT11 (Pin Change Interrupt 11)
PC2	ADC2 (ADC Input Channel 2) PCINT10 (Pin Change Interrupt 10)
PC1	ADC1 (ADC Input Channel 1) PCINT9 (Pin Change Interrupt 9)
PC0	ADC0 (ADC Input Channel 0) PCINT8 (Pin Change Interrupt 8)

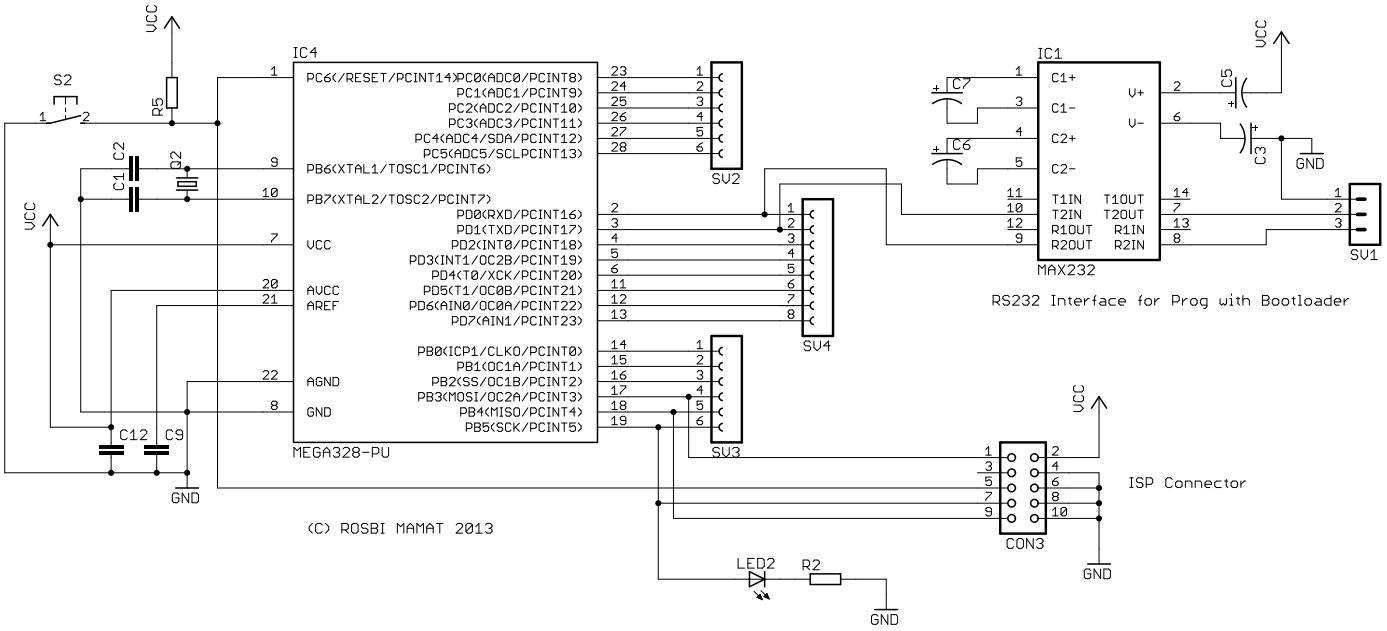
2.3 ATmega328 Pinout & Descriptions

- Port D (PD0-PD7) – serves as an 8-bit **bi-directional** digital I/O port with optional internal pull-ups. Port D pins are tri-stated when reset. Port D can be programmed to serve as **alternate function**:

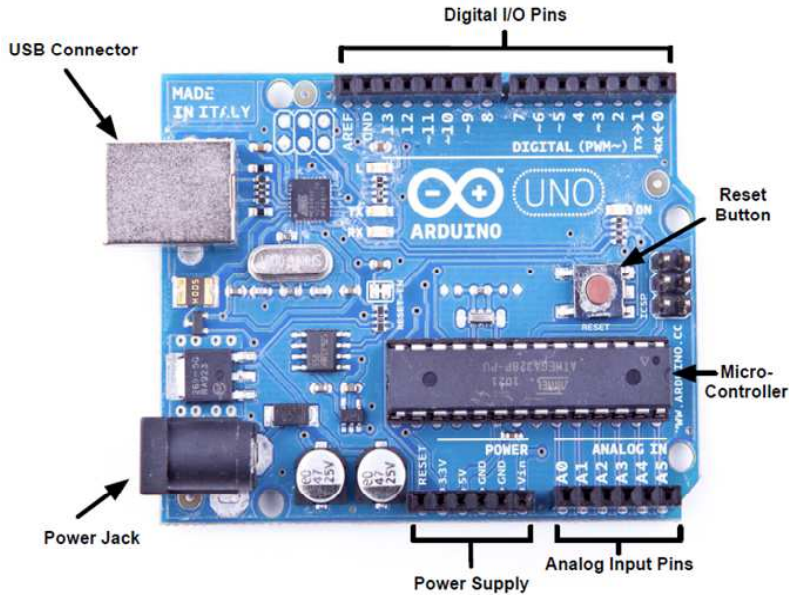
Port Pin	Alternate Function
PD7	AIN1 (Analog Comparator Negative Input) PCINT23 (Pin Change Interrupt 23)
PD6	AIN0 (Analog Comparator Positive Input) OC0A (Timer/Counter0 Output Compare Match A Output) PCINT22 (Pin Change Interrupt 22)
PD5	T1 (Timer/Counter 1 External Counter Input) OC0B (Timer/Counter0 Output Compare Match B Output) PCINT21 (Pin Change Interrupt 21)
PD4	XCK (USART External Clock Input/Output) T0 (Timer/Counter 0 External Counter Input) PCINT20 (Pin Change Interrupt 20)
PD3	INT1 (External Interrupt 1 Input) OC2B (Timer/Counter2 Output Compare Match B Output) PCINT19 (Pin Change Interrupt 19)
PD2	INT0 (External Interrupt 0 Input) PCINT18 (Pin Change Interrupt 18)
PD1	TXD (USART Output Pin) PCINT17 (Pin Change Interrupt 17)
PD0	RXD (USART Input Pin) PCINT16 (Pin Change Interrupt 16)

2.4 ATmega328 Hardware Design

- A schematic of an ATmega328 based system



2.5 Arduino Uno μc Board



- **Arduino Uno** – a μc board based on the ATMEL ATmega328 8-bit μc designed by Italian *Arduino* team & released under *common creative* license.

2.5 Arduino Uno μc Board – Specs.

- **Memory:** 31.5 KB **Flash** memory for storing **program** & 2 KB of RAM for storing **data**.
- **Digital I/O:** 14 pins can be used as **digital** input or output.
- **Analog I/O:** 6 pins can be used to read **analog** signals.
- **Software Dev. Tool:** can be programmed with the free **Arduino** software – can be downloaded from `www.arduino.cc`.
- Uno can communicate with a PC through the USB connector.

2.5 Arduino Uno – Setting Up

- **Hands on Exercise** – Refer to `www.arduino.cc`
  - Get an Arduino board and USB cable
  - Download & Install the Arduino IDE
  - Connect the board
  - Install the drivers
  - Launch the Arduino IDE
  - Open the blink example
  - Select your board
  - Select your serial port
  - Upload the program & run