

# Internal Peripherals of AVR MCUs

xBoard MINI™ Documentation

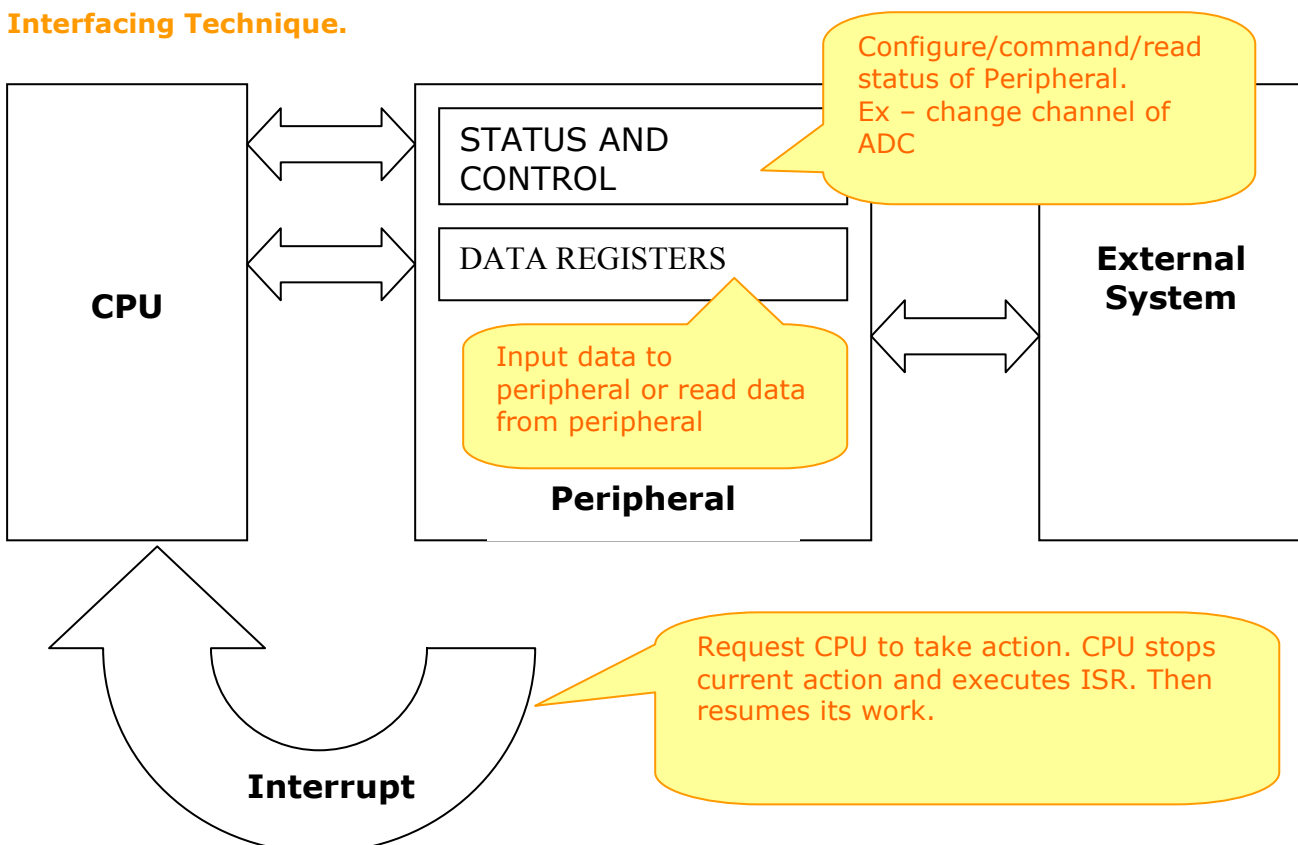
Each AVR MCU has several internal peripherals that give powerful ability to your projects. For example internal ADC can be used to convert analog value (say voltage output of some sensor) to a digital value that you can use. And a USART(Universal Synchronous Asynchronous Receiver Transmitter) can be used to connect your MCU with PC. (Example use - A PC serial mouse or A PC controlled Toy Rocket Launcher). This tutorial gives you overview of the process used to interface with these peripherals.

## Internal peripherals.

Now you know the basics of peripherals, lets see what peripherals are available in ATmega16 MCU.

- Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
- One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
- Real Time Counter with Separate Oscillator
- Four PWM Channels
- 8-channel, 10-bit ADC
- 8 Single-ended Channels
- 7 Differential Channels in TQFP Package Only
- 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
- Byte-oriented Two-wire Serial Interface
- Programmable Serial USART
- Master/Slave SPI Serial Interface
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator

## Interfacing Technique.



## Use of Internal peripheral of AVRs

A peripheral is connected to MCU by some special registers. The different registers of any peripherals can be logically of two types-

**1)Data registers** - which usually contains some data say byte received from USART or data just read from internal EEPROM.

**2)STATUS and CONTROL register** – these are used to configure the peripheral according to your need, and to control them from your program. For example the ADMUX register has following bits.

|              |              |              |             |             |             |             |             |
|--------------|--------------|--------------|-------------|-------------|-------------|-------------|-------------|
| <b>REFS1</b> | <b>REFS0</b> | <b>ADLAR</b> | <b>MUX4</b> | <b>MUX3</b> | <b>MUX2</b> | <b>MUX1</b> | <b>MUX0</b> |
|--------------|--------------|--------------|-------------|-------------|-------------|-------------|-------------|

The ADC (analog to digital convertor) has 8 input channels. Any one these can be selected by setting the MUX2-MUX0. To select ADC channel 0 set them to 000. Alternatively, to select channel 4 set them to 100 and so on.

### **Interrupts.**

As we saw the you can use the STATUS registers to see the what the peripheral has to say to the CPU, like it has just completed the conversion Analog value to digital and the data is ready. But in this way you have to constantly poll the registers. But one more advance feature is INTERRUPTs. Where the peripheral halts the CPU whenever it has something interesting. Then the CPU will jumps to a function called ISR (interrupt service routine) which is provided by you.

You can configure which events should interrupt the CPU and also provide the ISR.

### **Steps to use peripherals.**

- First setup the peripheral according to your need.
- Set interrupts and ISRs (Optional)(beginners may skip for simplicity).
- command the peripheral / read write data.

### **Whats next**

We will use a very useful peripheral of AVR – The ADC (analog to digital converter).