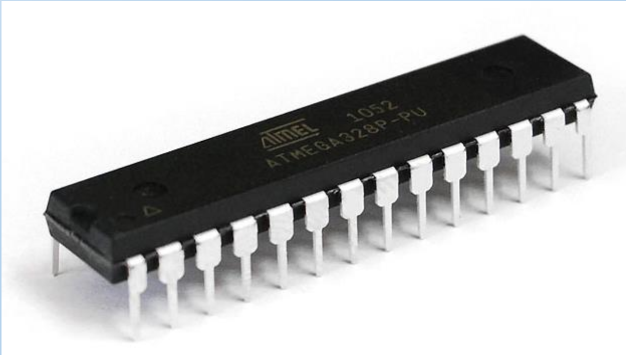
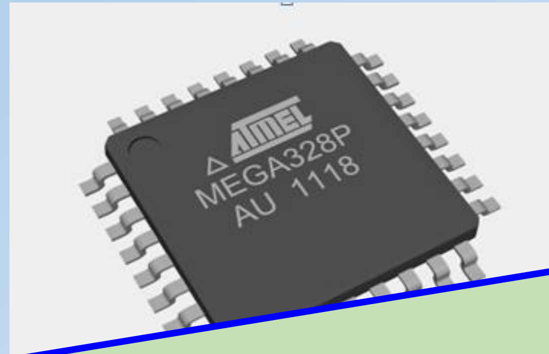
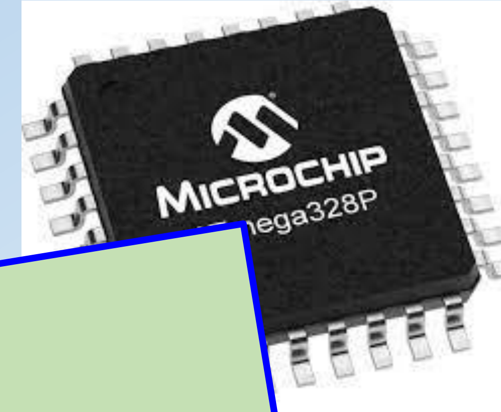
Vishwakarma Institute of Technology

FY - Department of Engineering, Sciences and Humanities



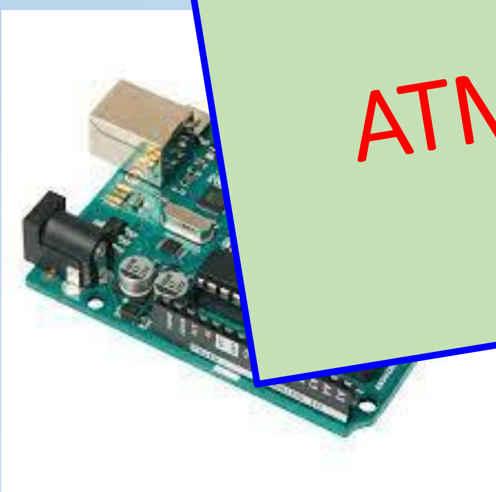


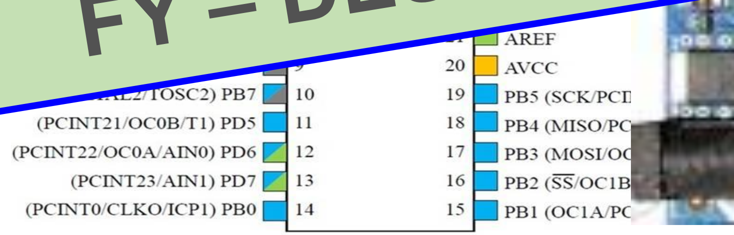


Micro Controller 3.3

ATMega328P - Interrupts

FY – DESH – VIT





Interrupts:-----

Methods to read / monitor the inputs ....

1) Polling = keep a watch all the time

2) Interrupt = work ONLY when interrupted

Vishwakarma Institute of Technology
FY - Department of Engineering, Sciences and Humanities

Method to read the input :-

1) Polling :-

Repeatedly reads a data port and tests the input data value.

The processor takes a measurement from a port/sensor at regular time intervals.

This happens, even if there is no change in the connected device.

Guaranteed or predictable time slots are allocated to each device that is being polled.

Polling is ...

- Simple to code.
- Not very efficient as the polling happens even if the sensor reading has not changed.

Vishwakarma Institute of Technology
FY - Department of Engineering, Sciences and Humanities

Method to read the input :-

Vishwakarma Institute of Technology
FY - Department of Engineering, Sciences and Humanities

2) Interrupts :-

Interrupt is a **signal generated by hardware or software** that triggers the **interrupt service routine - ISR** to run.

- It occurs only when needed.
- Interrupts are more efficient than polling but the timing of occurance is less predictable.
- More difficult to code than polling.

What happens when an interrupt occurs :-

Vishwakarma Institute of Technology
FY - Department of Engineering, Sciences and Humanities

1) The MuC completes the current instruction being executed and saves the address of the next instruction (PC) on the stack.

2) MuC saves the current status of all the interrupts internally.

3) It jumps to the memory location of the IVT (interrupt vector table) that contains the address of the ISR – (Interrupts service routine)

4) The MuC gets the address of the ISR from the IVT and jumps to it. It executes the ISR. The last instruction is RETI (return from interrupt).

5) The MuC returns to the location where it was interrupted. It gets the program counter (PC) address from the stack and starts execution from that address onwards……

Interrupts :-

Timer based Interrupts :-

Most processors have built-in timers which can be used to trigger processes when the timer interval ends. (Timer overflow)

Interrupts allow program ….

- ✓ **to respond to events when they occur and**
- ✓ **to ignore events until they occur**

Interrupts :-

Interrupts are used in following cases in ATMega328P -

1) **To detect pin changes (e.g. rotary encoders, button presses)**
2) **Watchdog timer (e.g. if nothing happens after 8 seconds, interrupt)**
3) **Timer interrupts – used for comparing/overflowing timers**
4) **SPI data transfers**
5) **I2C data transfers**
6) **USART data transfers**
7) **ADC conversions (analog to digital)**
8) **EEPROM ready for use**
9) **Flash memory ready**
10) **Power failure**
11) **Some abnormal mathematical exceptions** … and many more …..

Vishwakarma Institute of Technology
FY - Department of Engineering, Sciences and Humanities

Interrupts :-

Interrupts are useful for making things happen automatically in MuC programs, and can help solve timing problems as in Polling method.

Simple tasks for using an interrupt include -- (**To detect pin changes**)

e.g.

- Reading a rotary encoder
- Read a sound sensor that is trying to catch a sound of click
- Read an infrared sensor (photo-interrupter)
- Read key press in a 16 key keypad etc.

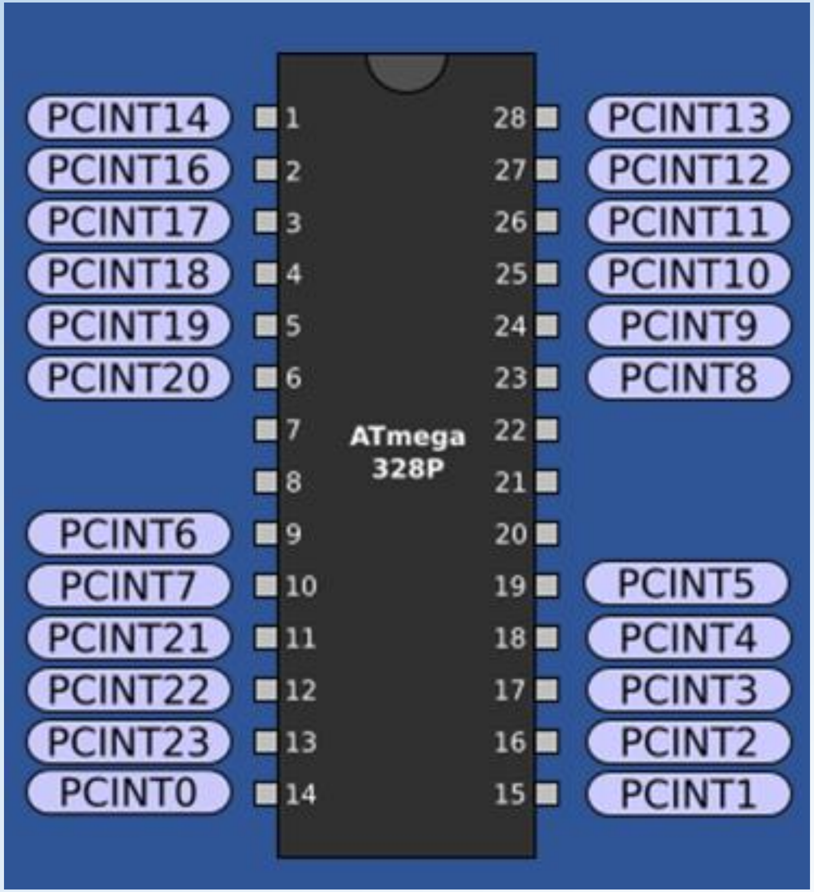
Thus, an interrupt can relieve the MuC to get some other work done and still not missing the input.

PCINT - Pin Change Interrupt

Vishwakarma Institute of Technology
FY - Department of Engineering, Sciences and Humanities

1) Reading the state of an input pin is necessary in microcontroller programs. (whether pin is 1 or 0)
2) **This is done by using Pin Change INTerrupts.**
3) Each of the digital I/O pins can be configured as PCINT by writing into interrupt registers in software.

The names and locations of each of the pin change interrupt pins.



	INT - Interrupt	PCINT - Pin Change Interrupt
1	INT refers to the dedicated hardware interrupt pins	PCINT refers to the interrupts that can be generated by almost all of the I/O pins. PCINT0-PCINT23
2	Each interrupt has a specific dedicated Vector address.	Group of pins share the same PCINT vector address. There are 3 such PCINT groups.
3	The INT pin is linked to a dedicated interrupt vector. Thus which pin has caused the interrupt can be found out.	Programmer needs to determine which pin change causes the interrupt within the ISR before acting on it. (As a group of pins share the same vector address)

Pin Change Interrupt Registers :-

Atmega 328 has 3 special registers associated with PCINT

1) **PC ICR** → Pin Change Interrupt Control Register

2) **PC IFR** → Pin Change Interrupt Flag Register

3) **PC MSKx** → Pin Change Mask Register.

(X - indicates the 3 PCINT groups 0, 1 and 2)

Pin Change Interrupt Control Register (PCICR)

1

Used to enable pin change interrupt on a pin.

-	-	-	-	-	PCIE2	PCIE1	PCIE0
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Group control Bit	PCINT group pins	Pin numbers On ATmega 328	Pin numbers On Arduino board
PCIE0	PCINT0 to PCINT7 (8 pins)	Port B0 to B7 8 pins	6 digital pins -- pins 8 to 13 excludes crystal pins
PCIE1	PCINT8 to PCINT14 (7 pins)	Port C0 to C6 7 pins.	6 pins -- A0 to A5 excludes Reset pin
PCIE2	PCINT16 to PCINT23 (8 pins)	Port D0 to D7 8 pins	8 pins – Digital pins 0 to 7

Set (1) = Enable and Clear (0) = Disable

Pin Change Interrupt Flag Register (PCIFR)

2

When interrupt is triggered by a pin, then corresponding group flag bit will be set.

					PCIF2	PCIF1	PCIF0
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Set (1) = Interrupt received and
Clear (0) = Interrupt not received

Once the PCINT group is activated in PCICR, next step is to select which pins of that group can trigger the interrupt.

Vishwakarma Institute of Technology
FY - Department of Engineering, Sciences and Humanities

To enable / mask a pin, PCMSK – (Pin Change Mask register) is used

Three PCMSK registers --- one for each group.

Each bit in the PCMSK register corresponds to a PCINT pin

3

PCMSK0 for pins PCINT 0 to 7

PCMSK1 for pins PCINT 8 to 14

PCMSK2 for pins PCINT 16 to 23

Set (1) = Enable

Clear (0) = Disable

Bit	7	6	5	4	3	2	1	0
	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Pin Change Interrupt – ISR Vectors

What is the Interrupt Vector Table – IVT ?

It is a memory location where address of the ISR is written.

What is the Interrupt Service Routine – ISR ?

It is the group of instructions which would be executed when an Interrupt occurs. Address of ISR is written in IVT.

What is Interrupt Flag ?

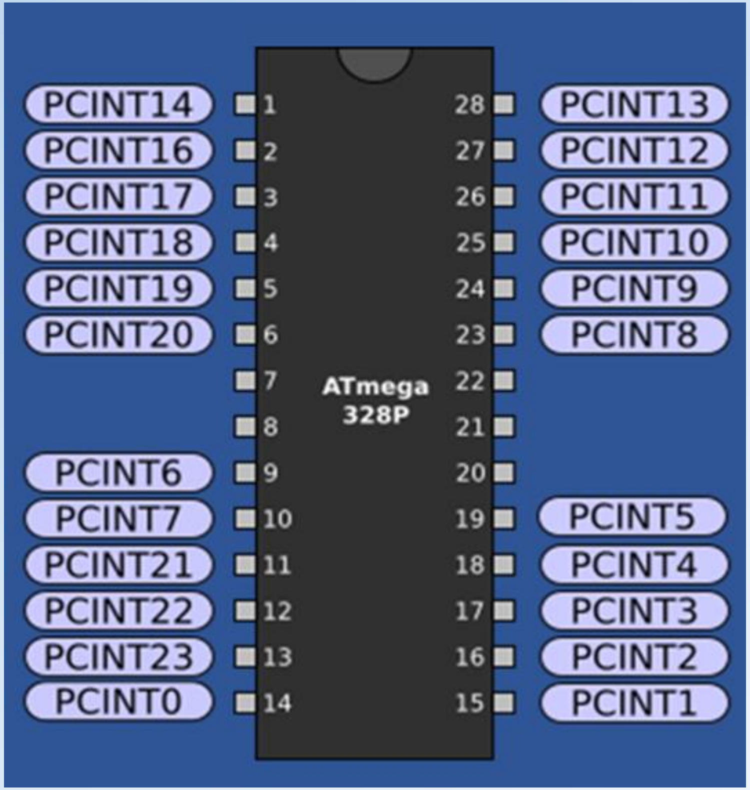
It is an indication showing that an interrupt has occurred. (Interrupt Flag is set when interrupt is acknowledged and cleared when either …

1) **execution of ISR starts OR**
2) **after ISR control returns to main program – programmer defined)**

Pin Change Interrupt – ISR Vectors

There are three vectors for the 3 groups (0, 1 or 2)

- *ISR (PCINT0_vect)* *// for pin group Port B*
- *ISR (PCINT1_vect)* *// for pin group Port C*
- *ISR (PCINT2_vect)* *// for pin group Port D*



Interrupt Vector Table (IVT) :-

Vector Number	Program Address	Interrupt definition	Vector name
1	0x0000	RESET External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset	RESET
2	0x0002	External Interrupt Request 0	INT0_vect
3	0x0004	External Interrupt Request 1	INT1_vect
4	**0x0006**	**Pin Change Interrupt Request 0**	**PCINT0_vect**
5	**0x0008**	**Pin Change Interrupt Request 1**	**PCINT1_vect**
6	**0x000A**	**Pin Change Interrupt Request 2**	**PCINT2_vect**
7	0x000C	Watchdog Time-out Interrupt	WDT_vect
8	0x000E	Timer/Counter2 Compare Match A	TIMER2_COMPA_vect

Types of Triggering :-

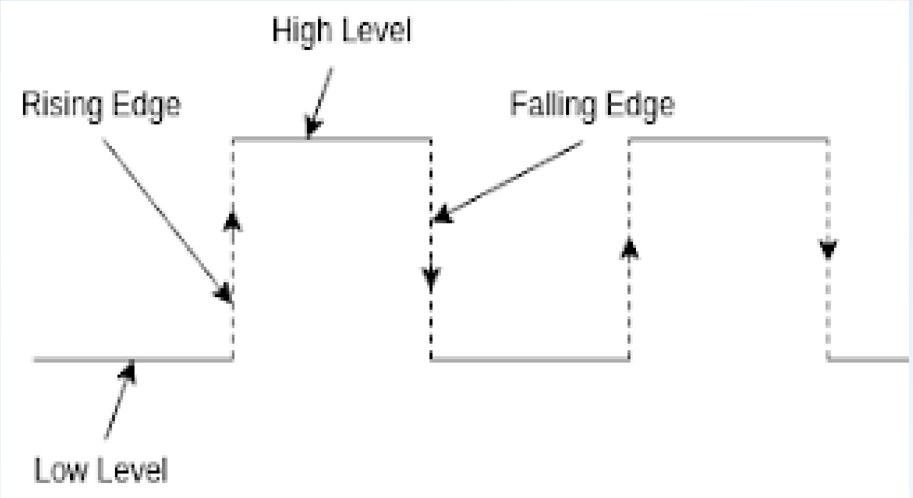
Trigger is what pushes the **electronic** devices from **no activity state** to **active state** with a small change in the input signal.

	Triggering	
No activity state	$\longrightarrow$	**Active state**

Types of Triggering :-

1) Edge triggering :-

2) Level triggering :-



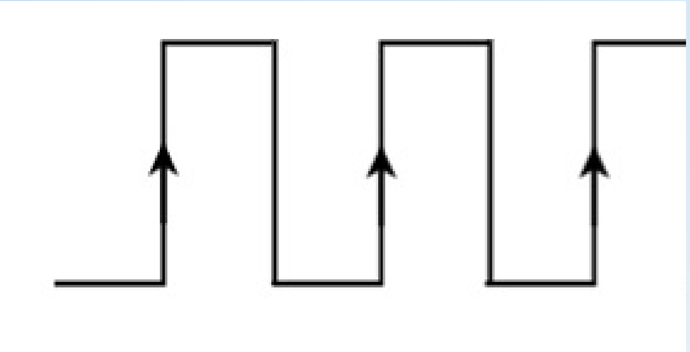
Vishwakarma Institute of Technology
FY - Department of Engineering, Sciences and Humanities

Vishwakarma Institute of Technology
FY - Department of Engineering, Sciences and Humanities

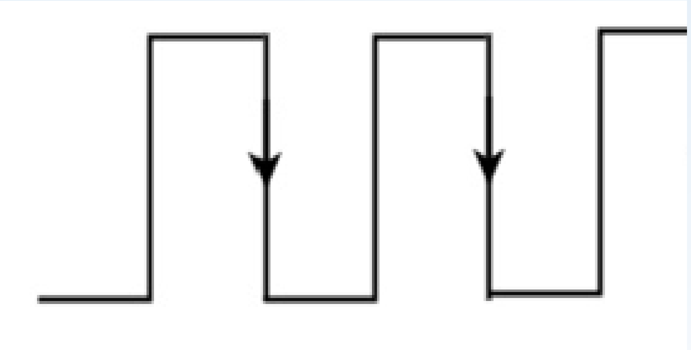
Types of Triggering :-

1) Edge triggering :- It is a type of triggering that allows a circuit to become active at the Rising (Positive) edge or the Falling (Negative) edge of the clock signal. Change is from Logic High to Logic Low (or vice versa)

a) **Positive edge triggering** :- If the circuit is operated with the clock signal that is transitioning from Logic Low to Logic High, then that type of triggering is known as **Positive edge triggering** (Rising edge triggering)



b) **Negative edge triggering :-** If the circuit is operated with the clock signal that is transitioning from Logic High to Logic Low, then that type of triggering is known as **Negative edge triggering** (Falling edge triggering)



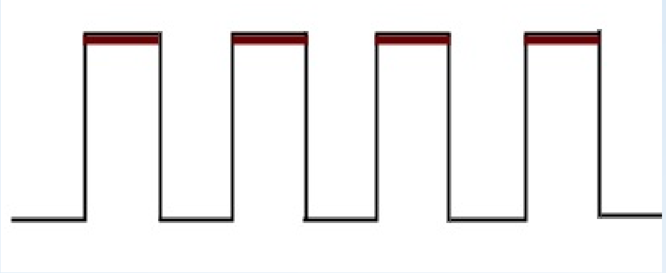
Types of Triggering :-

2) Level triggering :- It is a type of **triggering** that allows a circuit to become active when the clock pulse is on a particular **level**.

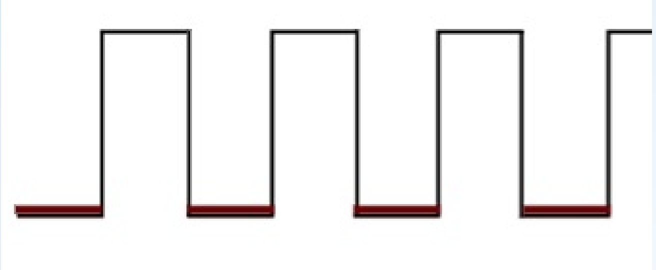
a) Logic High level i.e. Positive level triggering

b) Logic Low level i.e. Negative level triggering

a) **Positive level triggering :-** If the circuit is operated with the clock signal when it is in **Logic High**, then that type of triggering is known as **Positive level triggering**.

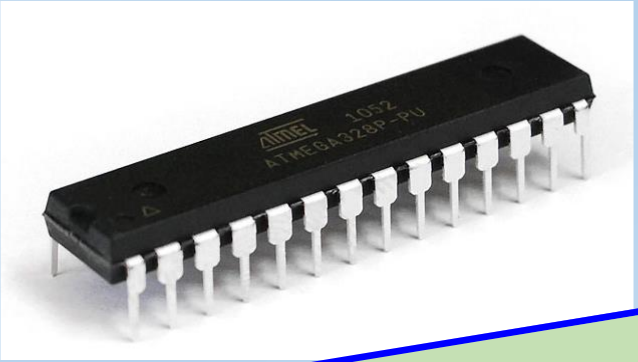


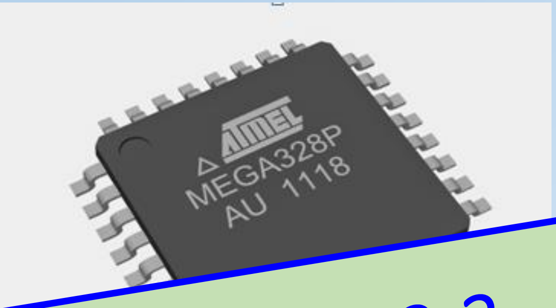
b) **Negative level triggering** :- If the circuit is operated with the clock signal when it is in **Logic Low**, then that type of triggering is known as **Negative level triggering**.

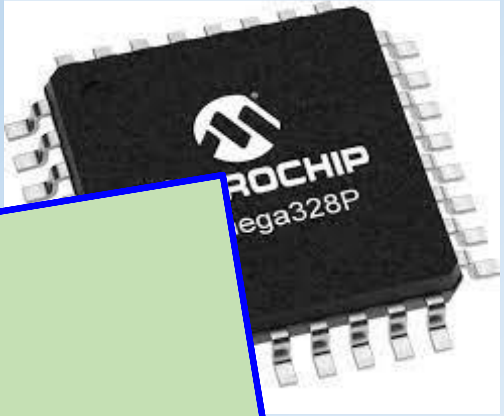


Vishwakarma Institute of Technology

FY - Department of Engineering, Sciences and Humanities







Micro Controller 3.3

ATMega328P Interrupts

Thanks !

FY – DESH – VIT