

Question 1: difference between Polling and Interrupt-driven?

The main difference between interrupt and polling is that in interrupt, the device notifies the CPU that it requires attention while, in polling, the CPU continuously checks the status of the devices to find whether they require attention. In brief, an interrupt is asynchronous whereas polling is synchronous.

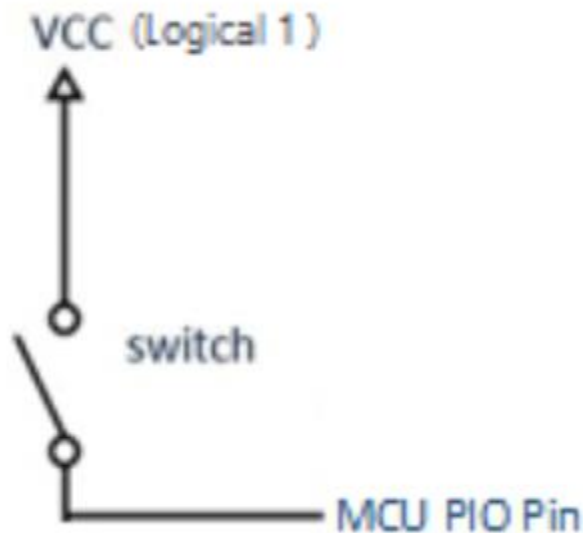
## I N T E R R U P T V E R S U S P O L L I N G

INTERRUPT	POLLING
An event that is triggered by external components other than the CPU that alerts the CPU to perform a certain action	An activity of sampling the status of an external device by a client program as a synchronous activity
When an interrupt occurred, the interrupt handler executes	In polling, the CPU provide the service
Can occur at any time	Occurs at regular intervals
Interrupt-request line indicates that device needs a service	Command ready bit indicates the device needs a service
Does not waste much CPU cycles	Wastes lot of CPU cycles
It is inefficient when the device interrupts the CPU frequently	It is inefficient in polling, when the CPU rarely finds requests from the devices
	Visit <a href="http://www.PEDIAA.com">www.PEDIAA.com</a>

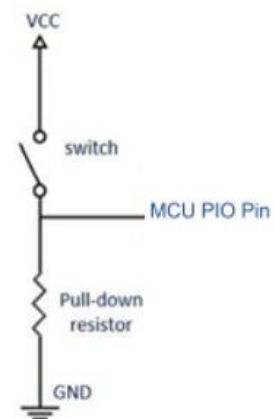
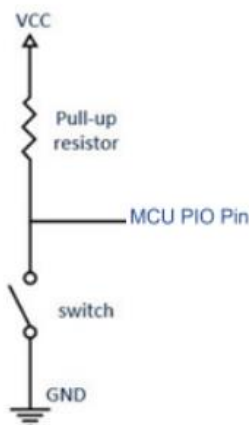
Source: <https://pediaa.com/what-is-the-difference-between-interrupt-and-polling/>

**پرسش:** چرا این روش برای فهمیدن اینکه چه زمانی کلید بسته شده درست نیست؟ در این مدار پایه میکرو در چه حالتی می باشد؟

using this method is incorrect because when the switch is open then pin is in floating state and the voltage there could be anywhere between 0 to 1. So the processor could think its 1 which is incorrect that is why we use pull-up and pull-down resistors.



**پرسش:** درباره چگونگی کارکرد مدار های بالا توضیح دهید. به چه دلیل نیاز به مقاومت (Pull-up/pull-Down) داریم؟



In the right picture when the switch is open because the **vcc** is not in the circuit, the only voltage creating current is the one between the pin headers and the ground which is so little that can be seen as 0, so the pins will have a value of 0 when the switch is open. When the switch is closed the pin header will have the voltage of **vcc**. This was using pull-down resistor.

In the left picture when the switch is closed the pin header has the voltage of ground so is a logical 0 and when the switch is open the pin header is connected to **vcc** through a resistor and since the current is little there won't be much voltage difference so the voltage of the pinheader could be seen as **vcc**.

**پرسش:** آیا رخ دادن یک اتفاق در صورت اعلام شدن (Assertion) لزوماً منجر به اجرای روال سرویس وقفه متناظر با آن می‌شود؟

No, the Arduino for example has one “nested vector interrupt” meaning if one or more interrupts happened when already another one is being processed they will be ignored.

**پرسش:** پایه‌های وقفه در برد ATmega 2560 و شیوه پیاده‌سازی وقفه ورودی را بدست آورید.

The pin headers used for interrupt in Arduino mega are 2,3,18,19,20 and 21(which should not be used when using the IIC protocol)

BOARD	DIGITAL PINS USABLE FOR INTERRUPTS
Uno, Nano, Mini, other 328-based	2, 3
Uno WiFi Rev.2, Nano Every	all digital pins
Mega, Mega2560, MegaADK	2, 3, 18, 19, 20, 21 (pins 20 & 21 are not available to use for interrupts while they are used for I2C communication)
Micro, Leonardo, other 32u4-based	0, 1, 2, 3, 7
Zero	all digital pins, except 4
MKR Family boards	0, 1, 4, 5, 6, 7, 8, 9, A1, A2
Nano 33 IoT	2, 3, 9, 10, 11, 13, A1, A5, A7
Nano 33 BLE, Nano 33 BLE Sense	all pins
Due	all digital pins
101	all digital pins (Only pins 2, 5, 7, 8, 10, 11, 12, 13 work with <b>CHANGE</b> )

Source: <https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/>

#### How to use:

##### Digital Pins With Interrupts

The first parameter to attachInterrupt() is an interrupt number. Normally you should use digitalPinToInterrupt(pin) to translate the actual digital pin to the specific interrupt number. For example, if you connect to pin 3, use digitalPinToInterrupt(3) as the first parameter to attachInterrupt().

**For more info you can use refer to the source explaining the complete process in more detail manner.**

**پرسش:** اگر بخواهیم در زمان تغییر مقدار پایه، وقفه فعال شود از چه mode ای درون تابع attachInterrupt استفاده می‌شود؟

the **CHANGE** state is used for this case.

**پرسش:** انواع اتفاق های ورودی را که واحد GPIO در برد آردوینو ATmega2560 می تواند رخ دادن آن ها را بفهمد و اعلام کند بنویسید.

It is said in the Arduino documentation:

mode: defines when the interrupt should be triggered. Four constants are predefined as valid values:

- **LOW** to trigger the interrupt whenever the pin is low,
- **CHANGE** to trigger the interrupt whenever the pin changes value
- **RISING** to trigger when the pin goes from low to high,
- **FALLING** for when the pin goes from high to low.

The Due, Zero and MKR1000 boards allow also:

- **HIGH** to trigger the interrupt whenever the pin is high.

Source: <https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/>

So the 5 states are: **LOW** , **CHANGE**, **RISING** , **FALLING** , **HIGH**