## **LAB1**

**Negin kheirmand**

**9831023**

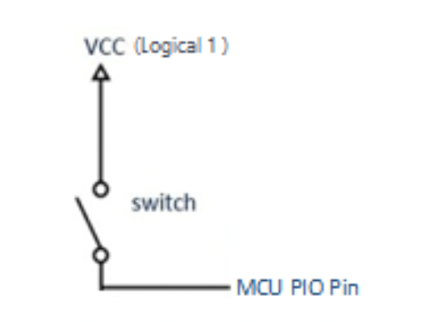
Question 1: difference between Polling and Interrupt-driven?

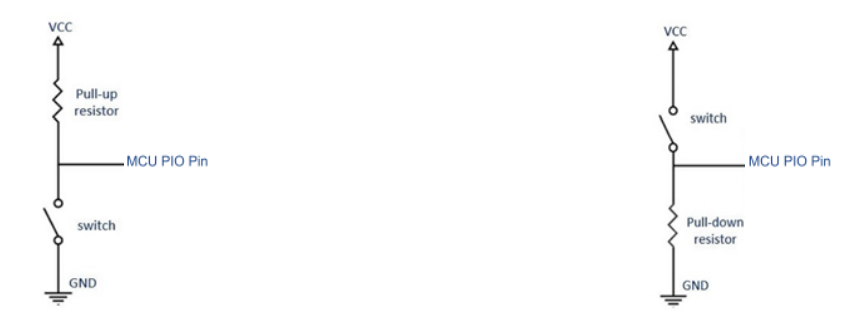
The main difference between interrupt and polling is that in interrupt, the device notifies the CPU that it requires attention while, in polling, the CPU continuously checks the status of the devices to find whether they require attention. In brief, an interrupt is asynchronous whereas polling is synchronous.



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

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.



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.

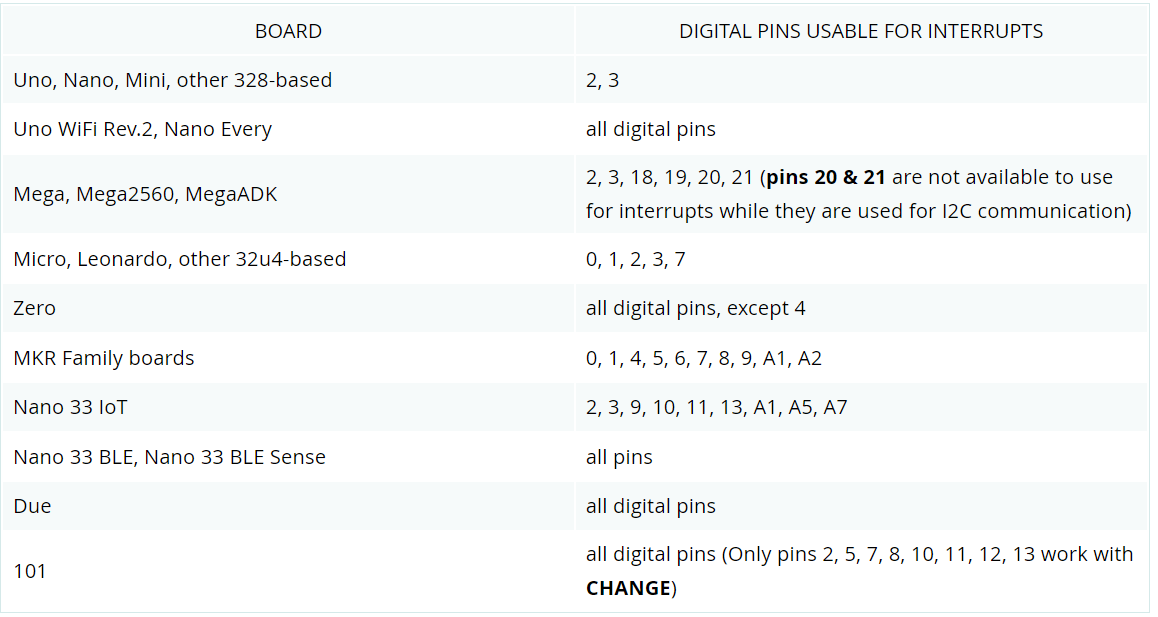
In the left picture when the switch is closed the pin header has the voltage of ground so is a logical 0 and when the switch is open the pin header is connected to vcc through a resistor and since the current is little there won’t be much voltage difference so the voltage of the pinheader could be seen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



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.



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)



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.**

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

****

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**