Polling

- Constantly reading a memory location, in order receive updates of an input value

```c
#include "system.h"
#include "io.h"

int main() {
    unsigned char key;
    unsigned char val = 0x01;
    IOWR( LED_BASE, 0, val );
    while(1) {
        key=IORD(KEY_BASE,0);
        if( key == 0x02 ){
            // key number 1 pressed
            // slide the bit one position to the left
            val = val << 1 | val >> 7;

            // update the LEDs
            IOWR( LED_BASE, 0, val );
            // wait until the key is released
            while( IORD( KEY_BASE != 0x03 ) ){}
        }
    }
    return 0;
}
```
Altera Interrupts

- Initialization

- Enable the interrupt for the specific input
  
  \[
  \text{IOWR_ALTERA_AVALONPIO_IRQ\_MASK(<BASE>, <MASK>)} \;
  \]

- Set the edge capability
  
  \[
  \text{IOWR_ALTERA_AVALONPIO\_EDGE\_CAP(<BASE>, <VAL>)} \;
  \]

- Register handler

  \[
  \text{alt\_ic\_isr\_register( <IRQ\_Controller\_ID>, <IRQ>, 
  <isr\_function>, <isr\_context>, <flags> )} \;
  \]
Tasks

- Control the LED with KEY on your board
- Try out and understand the Interrupt based KEY-LED package