

## Programming the Watchdog Timer

The mainboard is equipped with a watchdog timer that resets the CPU or generates an interrupt if processing comes to a standstill for whatever reason. This feature ensures system reliability in industrial stand-alone and unmanned environments.

# Programming the watchdog timer

---

## How to program the WATCHDOG TIMER

1. To set the time-out interval of watchdog timer:
  - output the desired value to port 0x443. Since the data is of 1 byte, the maximum value will be 255. In our design 2 ~ 255 will denote 2 ~ 255 sec.
  - `outportb(0x443, 30 <HEX>); // set watchdog to 30 seconds`
2. To set the time-out event:
  - output data to port 0x444,
  - `Timeout=SEC.Q=1`
  - `WDRST=IRQSET=3 & TIMEOUT`
  - `IRQ15=IRQSET=2 & TIMEOUT`
  - `IRQ11=IRQSET=4 & TIMEOUT`
  - `IRQ10=IRQSET=5 & TIMEOUT`
  
  - `outportb(0x444, 3); // set time-out event to reset-system`
3. To disable watchdog timer:
  - output value 0 to port 0x443
  - `outportb(0x443, 0); // disable watchdog timer`
4. To enable or refresh watchdog timer (the watchdog timer will return to its initial value, then count down):
  - access the I/O port 0x443, e.g.

```
outputb(0x443, data); // refresh watchdog timer
```

\* note: if you want to refresh the watchdog timer, you have to disable it first.

### Demo program

```
outputb(0x444, 3); // set time-out event to reset-system
outputb(0x443x 10); // set time-out interval to 10 seconds
customer_job(); // execute your job here, be sure your
                // job will finished within 10 seconds
outputb(0x443, 0); // refresh watchdog timer, otherwise
                  // the system will reset after time-out
outputb(0x443, 20); // set time-out interval to 20 seconds
another_job(); // another job finished in 20 seconds
outputb(0x443, 0) // disable watchdog timer
```

...

...