

Интернет вещей

Операционные
системы

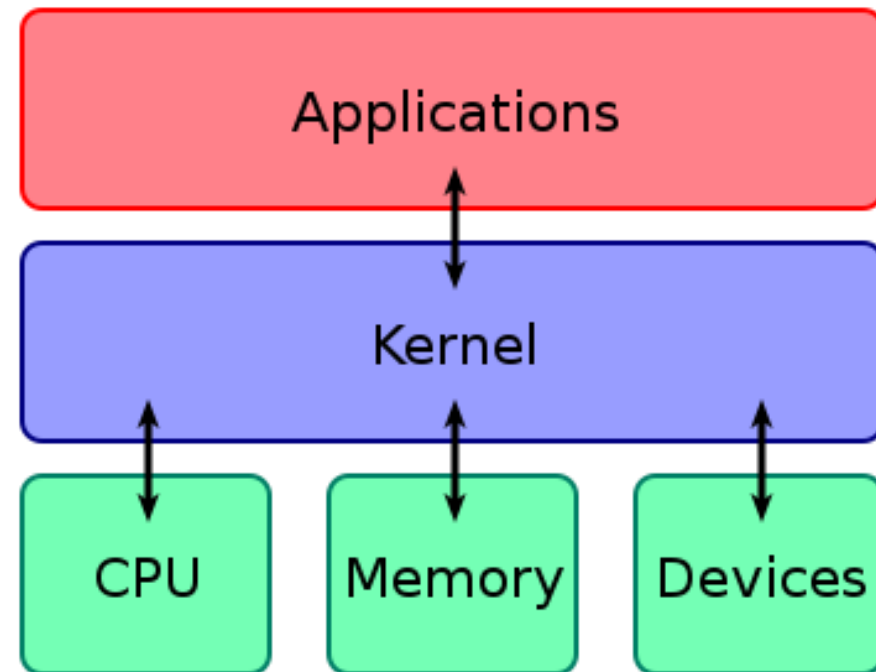
Последовательные

Операционная система

система управления ресурсами вычислительной системы

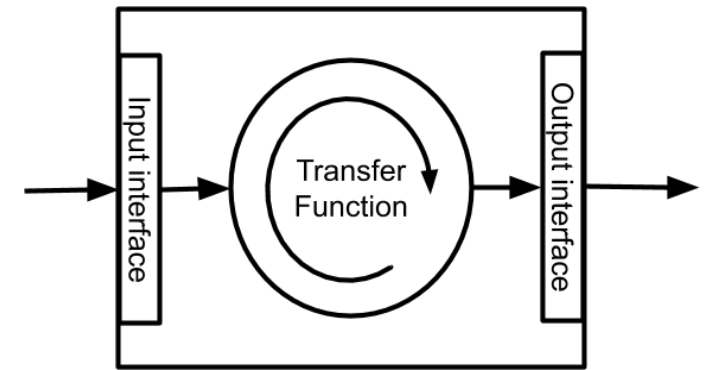
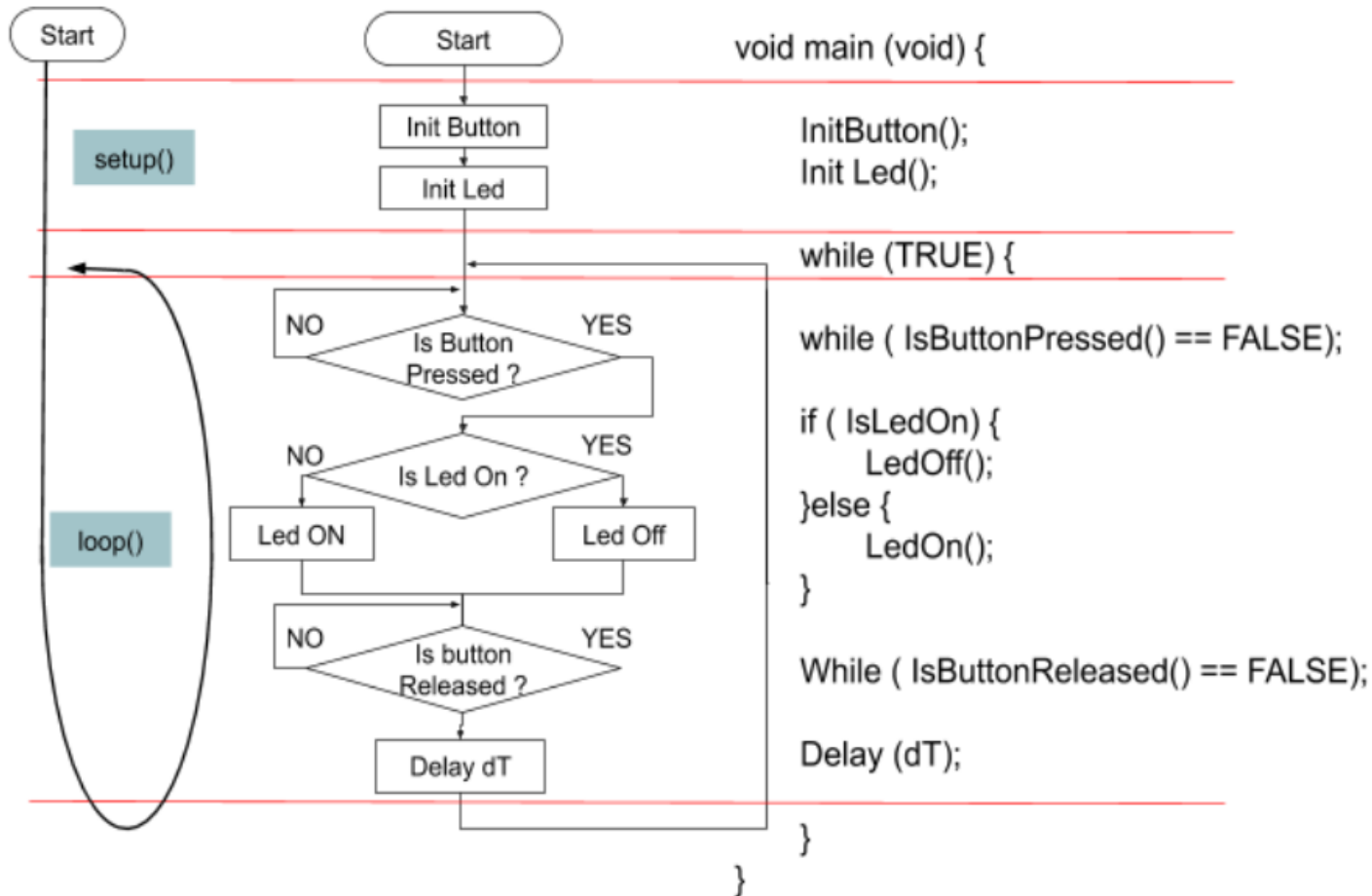
Управляемые ресурсы:

- Память
- Периферия
- Время обработки



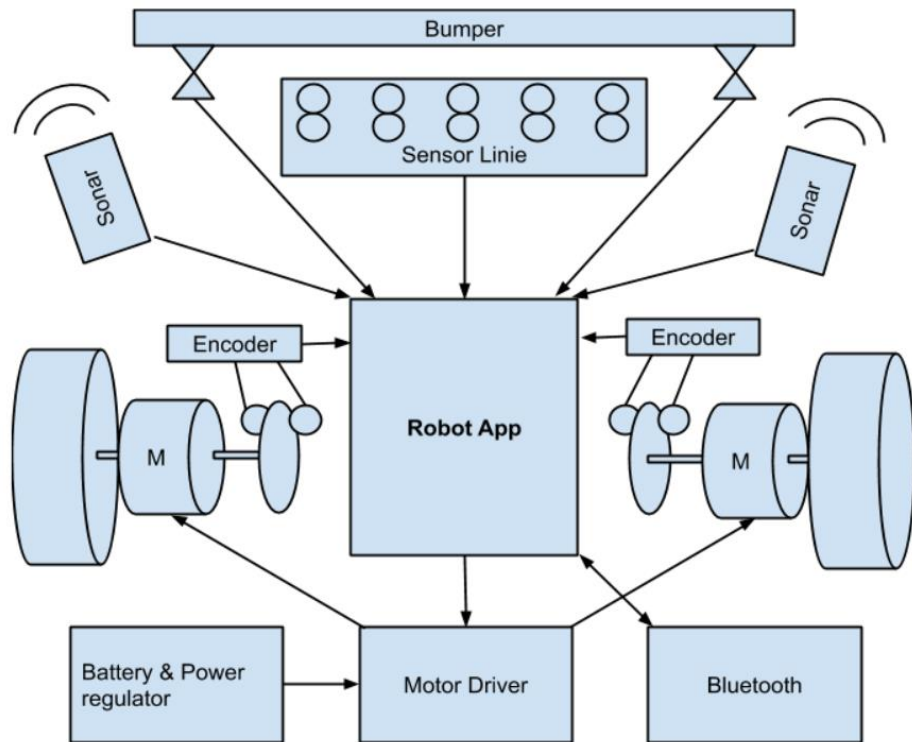
Single-process – Infinite loop

Классическое приложение– Buton / Led

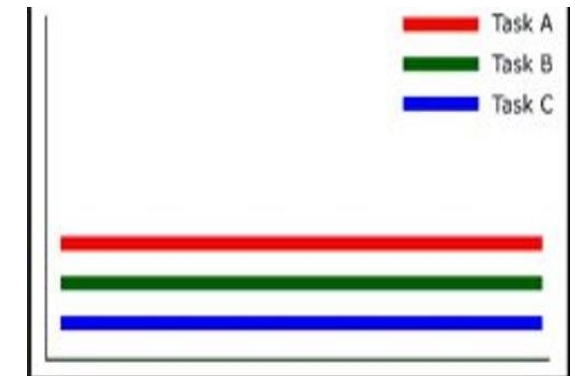
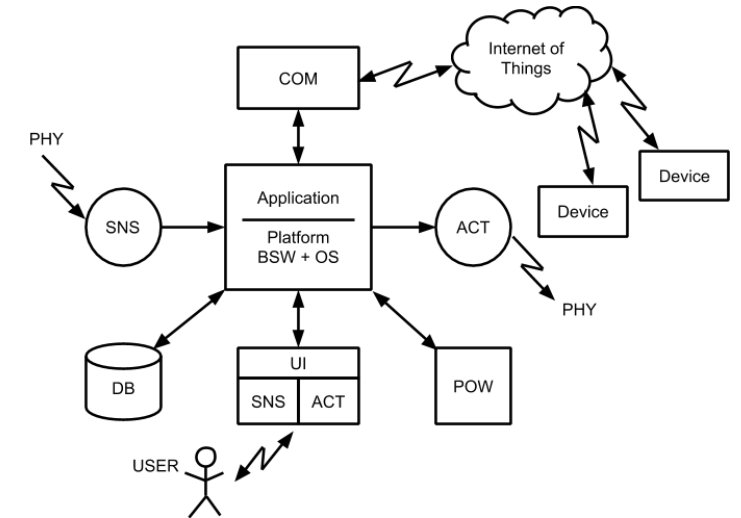


Multi tasking - Problem

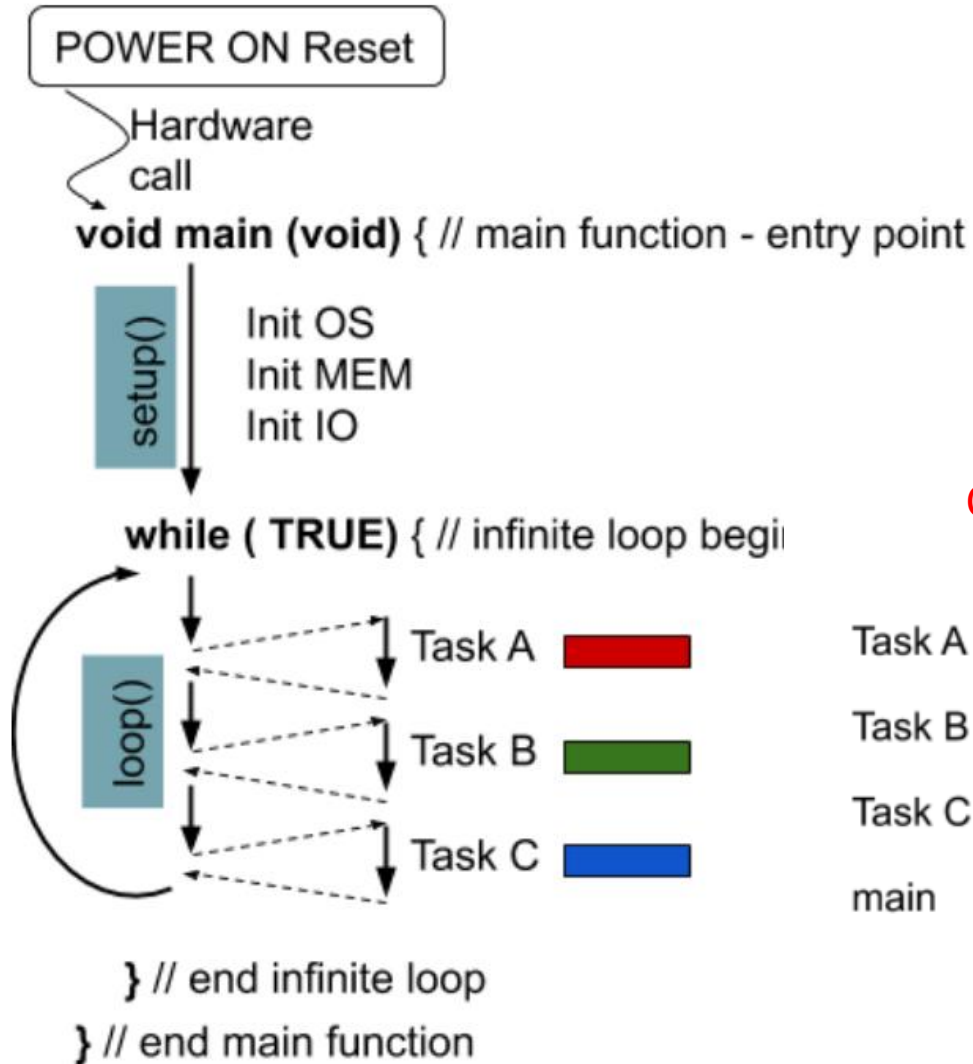
сосуществование нескольких функций одновременно



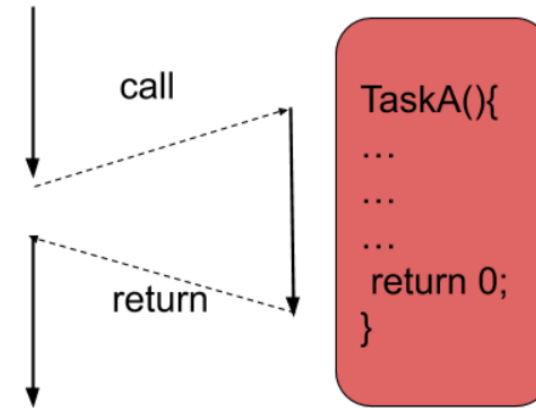
- Взаимодействие с пользователем
- Сенсор
- Привод
- Управление
- Коммуникация



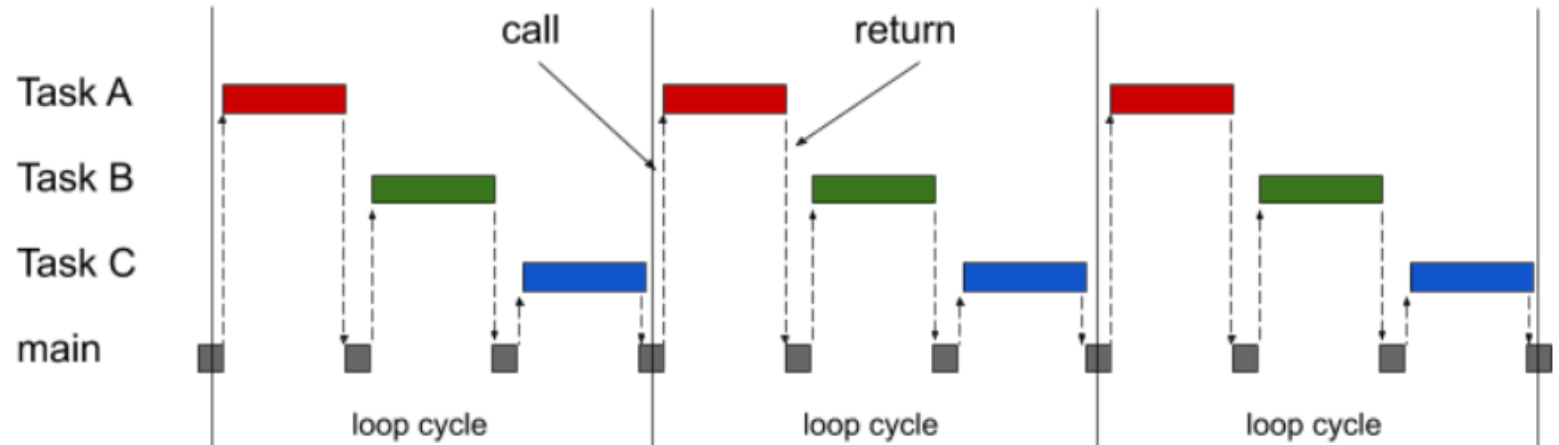
Multi tasking - Single process



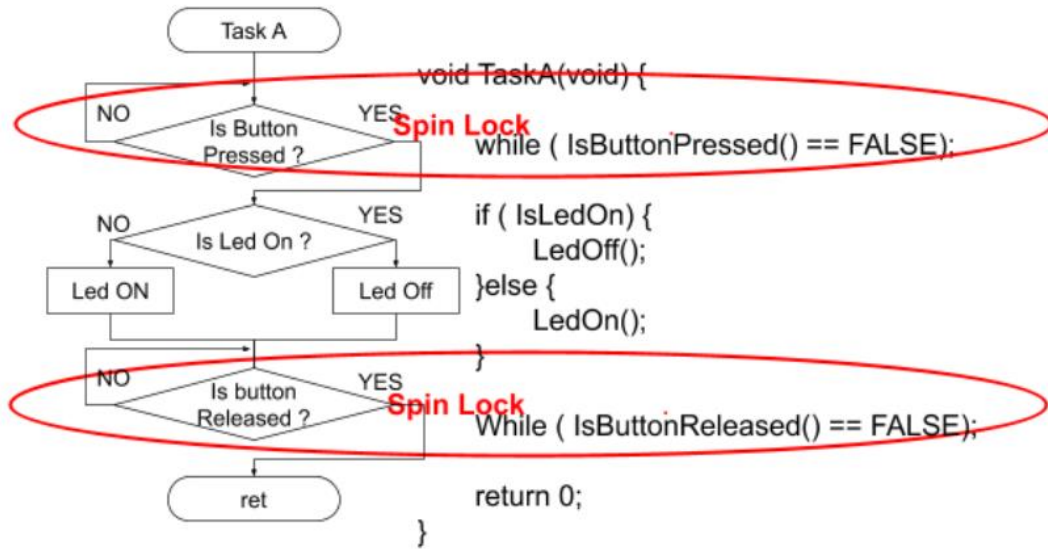
Task switch context



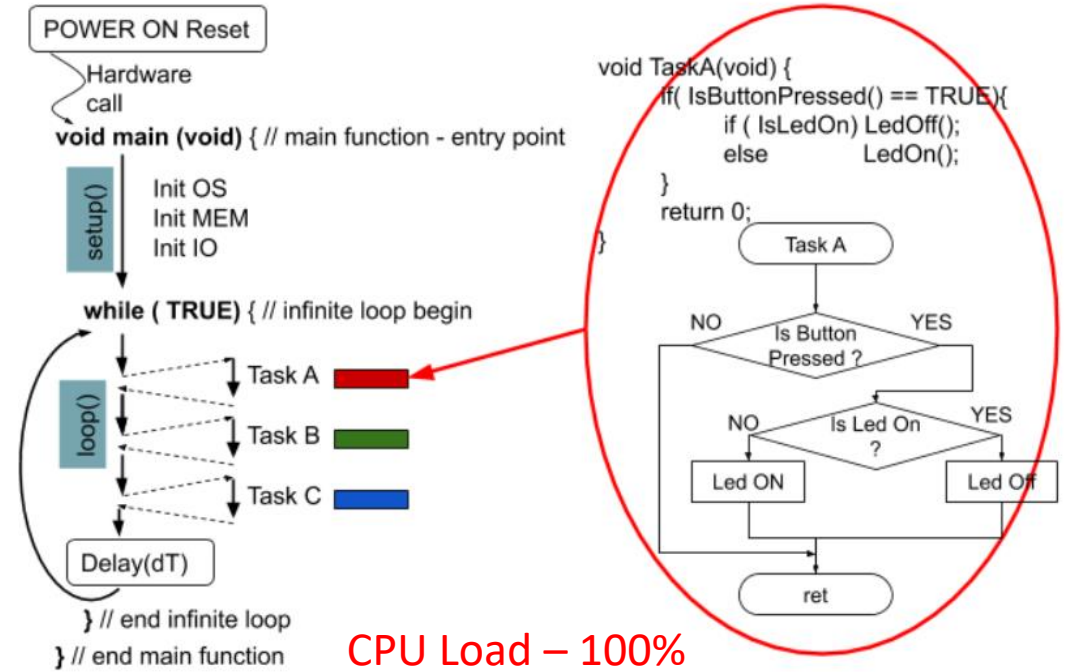
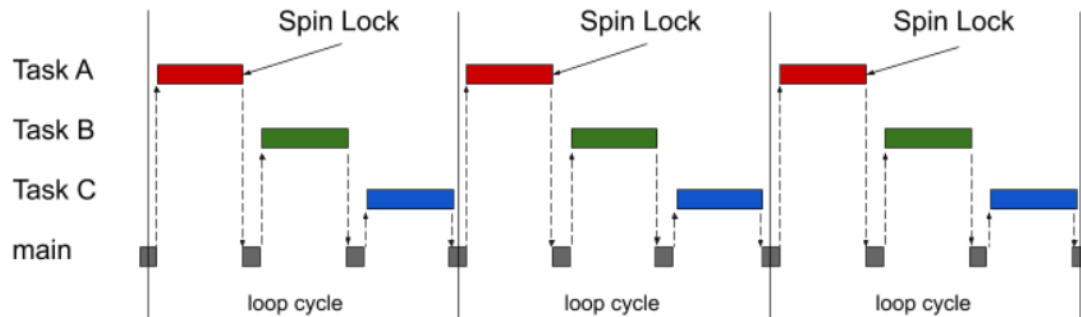
CPU Load – 100%



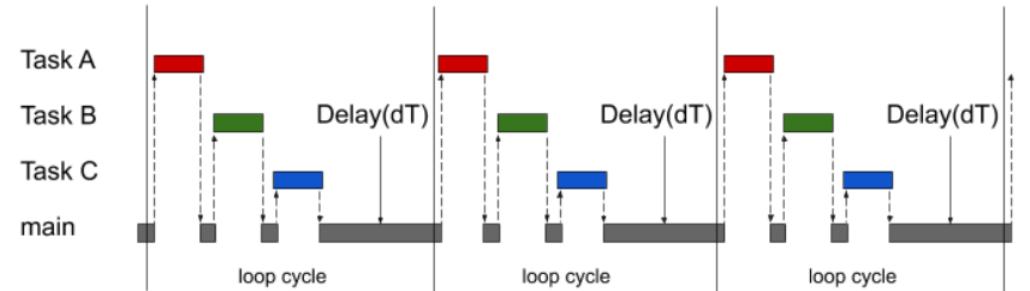
Multi tasking – Spin Lock to Sequential



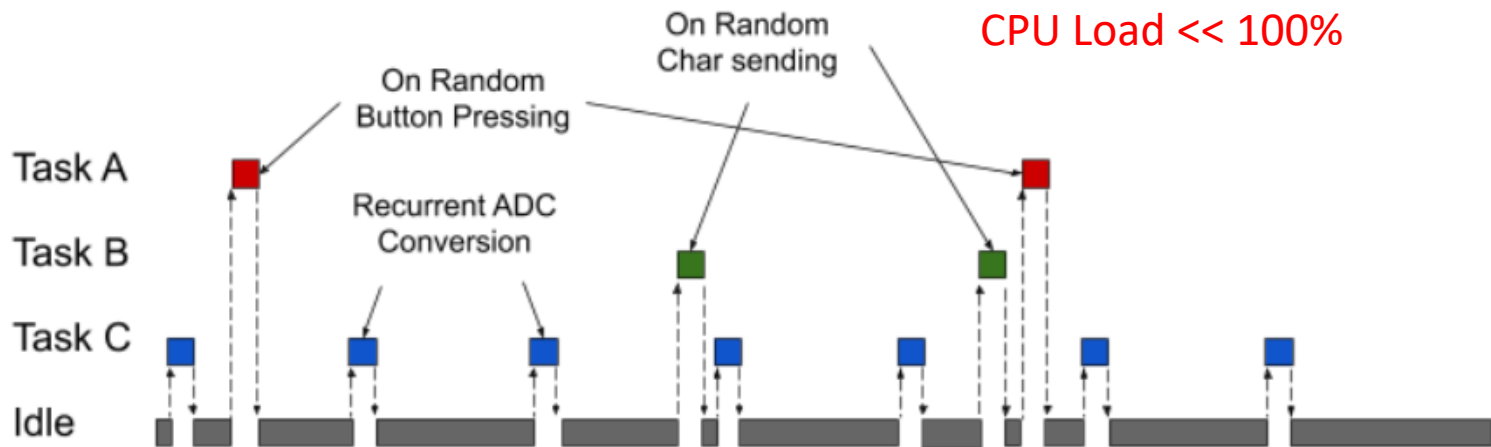
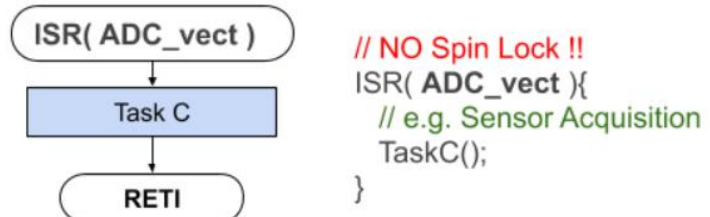
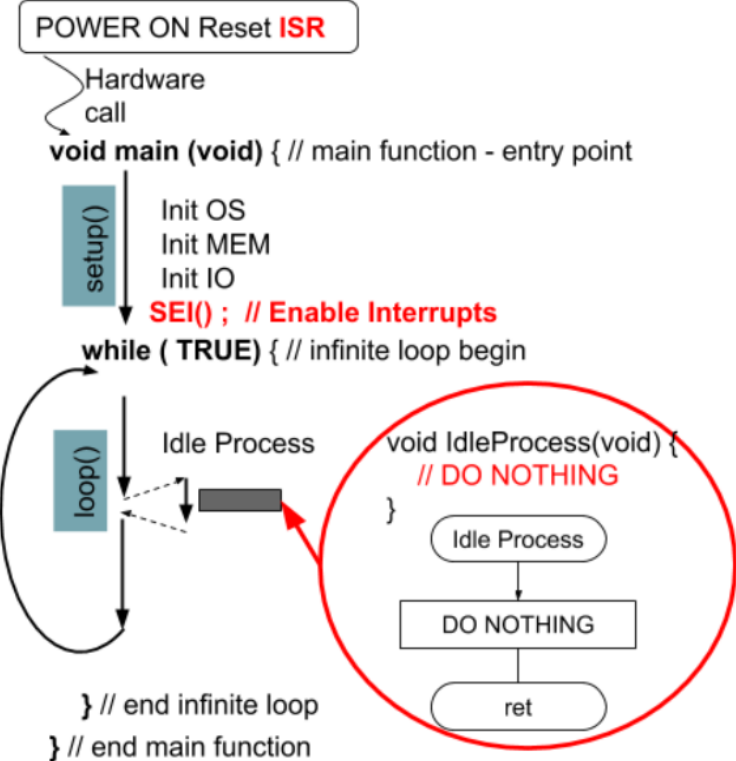
CPU Load – 100%



CPU Load – 100%



Multi tasking – Event driven & Interrupts



Multi tasking – Timer Interrupt based

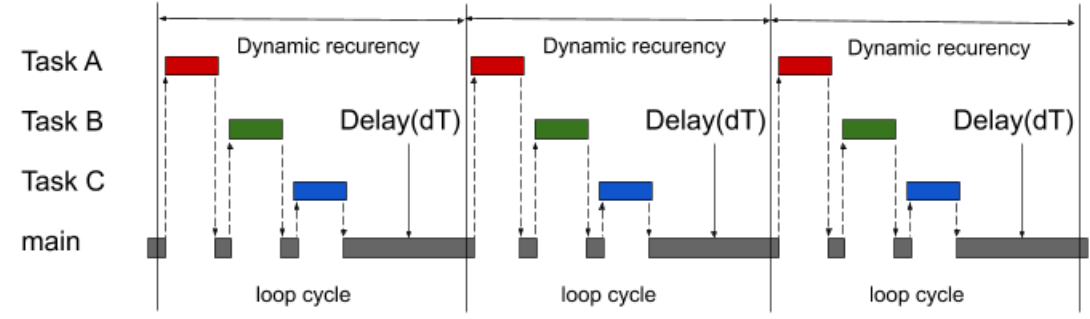
```

POWER ON Reset ISR
Hardware call
void main (void) { // main function - entry point
  setup()
  Init OS
  Init MEM
  Init IO
  SEI(); // Enable Interrupts
  while ( TRUE) { // infinite loop begin
    loop()
  } // end infinite loop
} // end main function

void IdleProcess(void) {
  // DO NOTHING
  Idle Process
  DO NOTHING
  ret
}

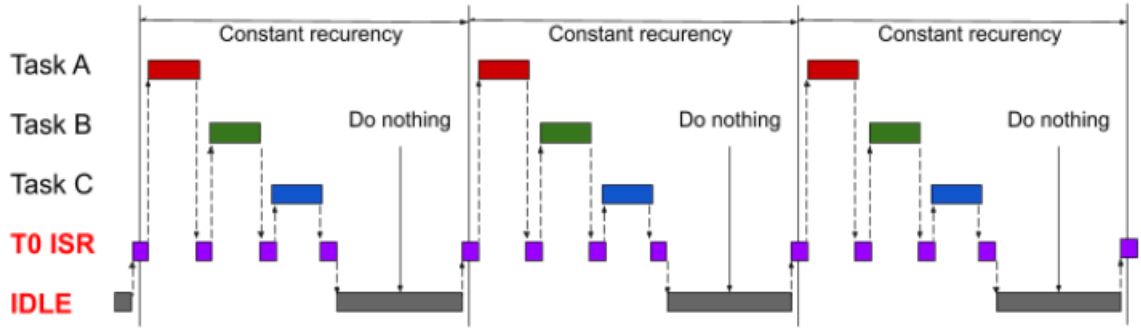
// NO Spin Lock !!
ISR( T0_OVF_vect ){ //e.g. 10 ms
  // e.g. Button / Led
  TaskA();
  // e.g. BT Command Interpreter
  TaskB();
  // e.g. Sensor Acquisition
  TaskC();
  RETI
}
  
```

CPU Load – 100%



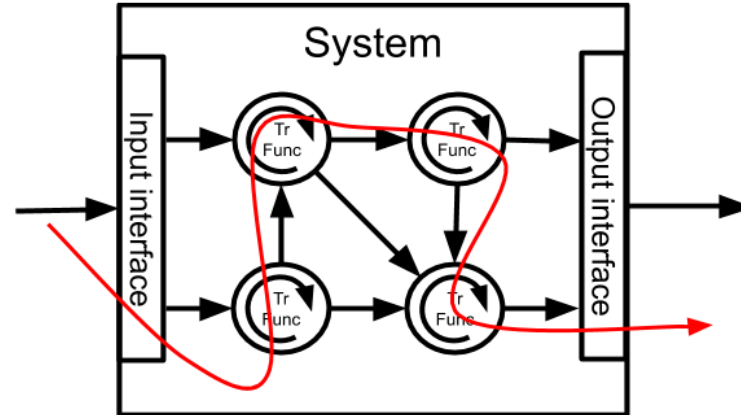
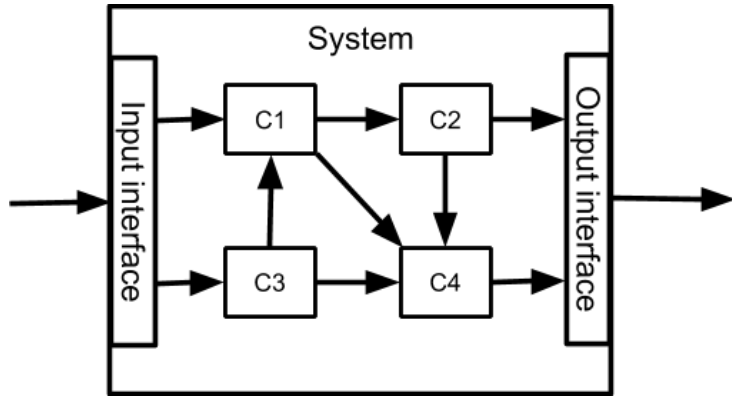
Multi tasking - Single process

CPU Load < 100%



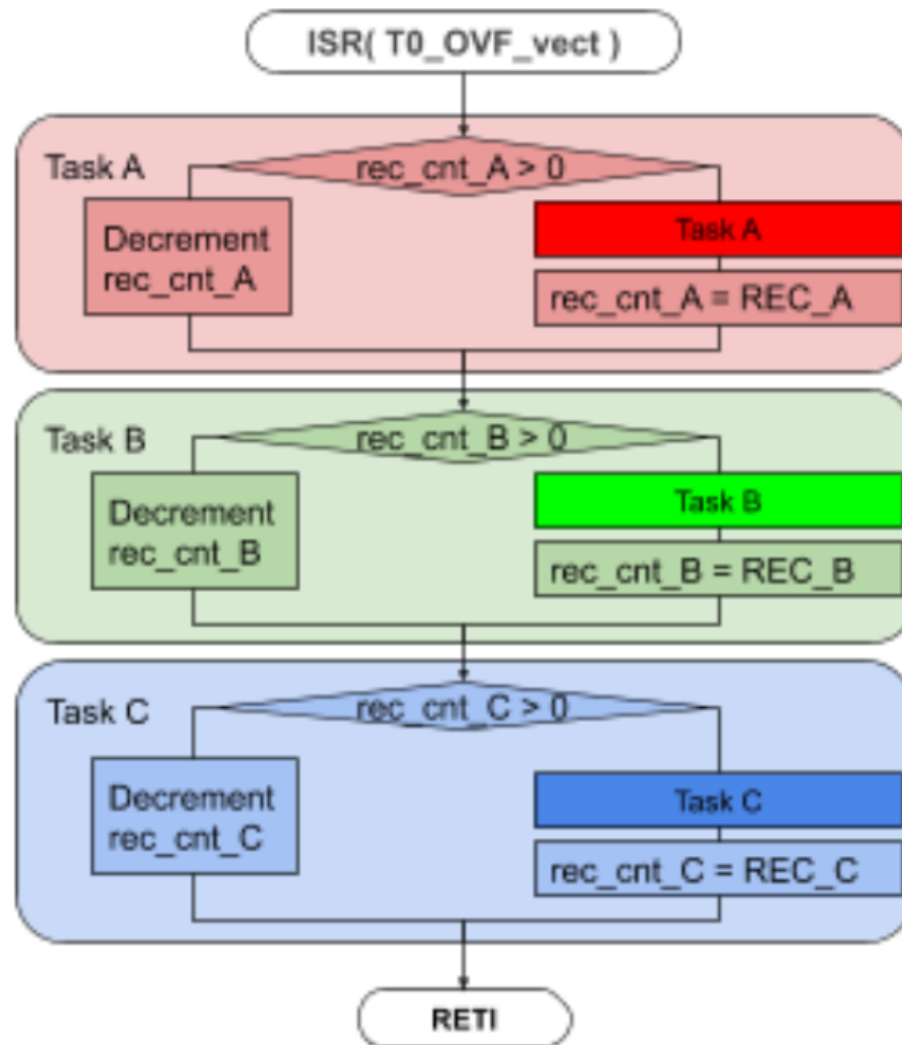
Multi tasking – Timer ISR based

Multi tasking – Task Recurrence & Order



1. C3 – Task 3: Recurrence = e.g. 1 ms , Offset 0 ms
2. C1 – Task 1: Recurrence = e.g. 2 ms , Offset 1 ms
3. C2 – Task 2: Recurrence = e.g. 3 ms , Offset 3 ms
4. C4 – Task 4: Recurrence = e.g. 5 ms , Offset 6 ms

Multi tasking – Task Recurrence & Order



Task Config

Task A: REC_A = 3 ms , OFST_A 3 ms

Task B: REC_B = 2 ms , OFST_B 1 ms

Task C: REC_C = 4 ms , OFST_C 0 ms

Recurrency Counters Init:

rec_cnt_A = OFST_A

rec_cnt_B = OFST_B

rec_cnt_C = OFST_C

```
ISR( T0_OVF_vect ) { // 1 ms
    // e.g. Button / Led
    if(--rec_cnt_A <= 0) {
        TaskA();
        rec_cnt_A = REC_A;
    }
    // e.g. BT Command Interpreter
    if(--rec_cnt_B <= 0) {
        TaskB();
        rec_cnt_B = REC_B;
    }
    // e.g. Sensor Acquisition
    if(--rec_cnt_C <= 0) {
        TaskC();
        rec_cnt_C = REC_C;
    }
}
```

Multi tasking – Task Distribution

