

# *ПО СРВ*

Тема лекции:

Объектно ориентированное моделирование  
систем реального времени(ROOM)

БГТУ им. Д.Ф. Устинова, каф. ИЗ

# *Структура лекции*

- Диаграмма состояний.
- Диаграмма композитной/составной структуры.
- Пример.
- Практические реализации:

# Диаграмма состояний

Диаграмма состояний - графическое представление конечного автомата.

Конечный автомат — описывает поведение отдельной системы(объекта) в форме последовательностей состояний.

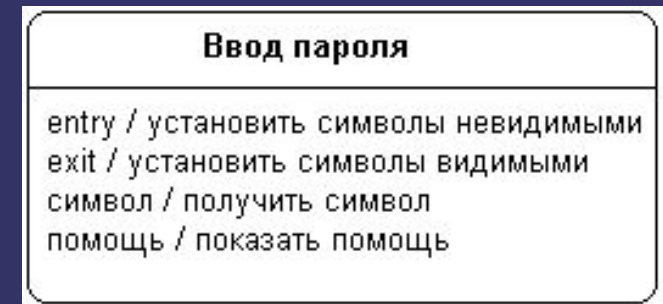
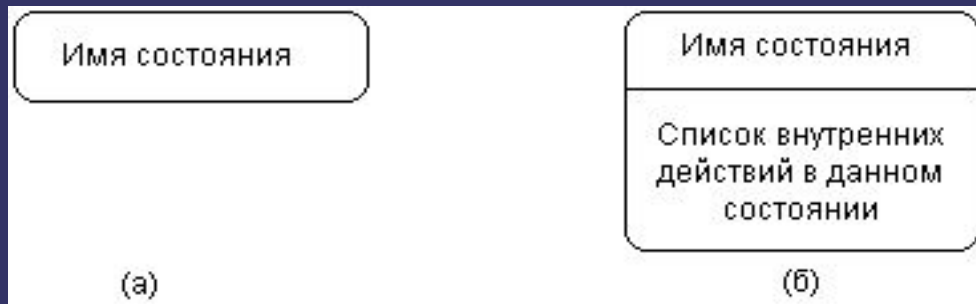


# Диаграмма состояний (*statechart diagram*), основные принципы

- Независимость состояний.
- Детерминированность...
  - Состояний
  - Переходов
- Разделенность переходов от состояний по временным характеристикам.
- Конечность количества состояний.
- Связанность структуры. (изолированные участки, тупиковые состояния).

# Состояние.

- Классическое представление.
- UML описание.



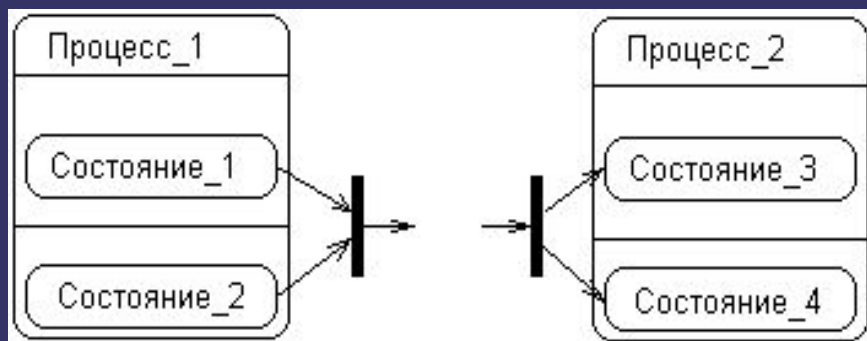
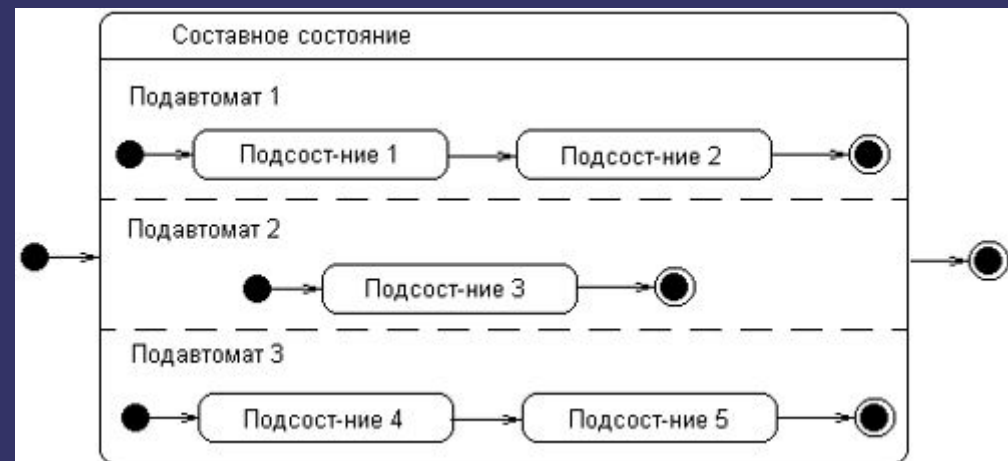
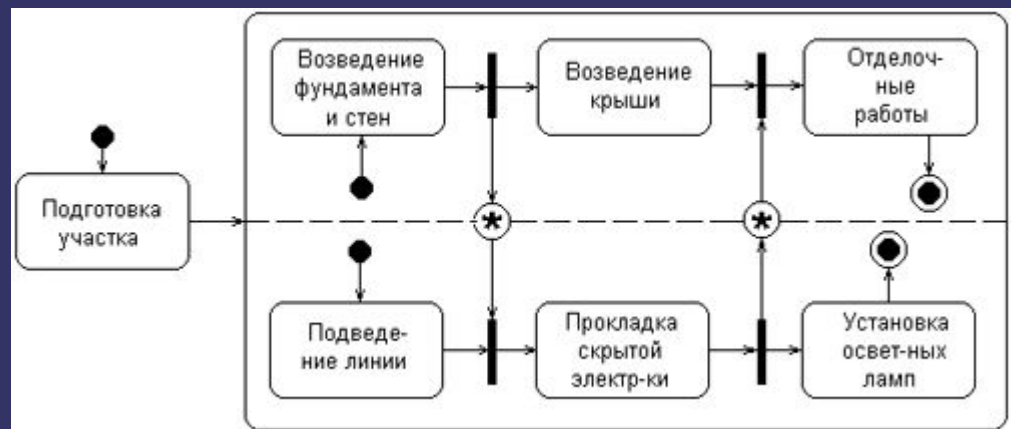
# Переход

Переход представляет отношение между 2-мя состояниями, указывает факт смены одного состояния на другое.

- Событие
- Сторожевое условие
- Действие перехода



# Специальные состояния

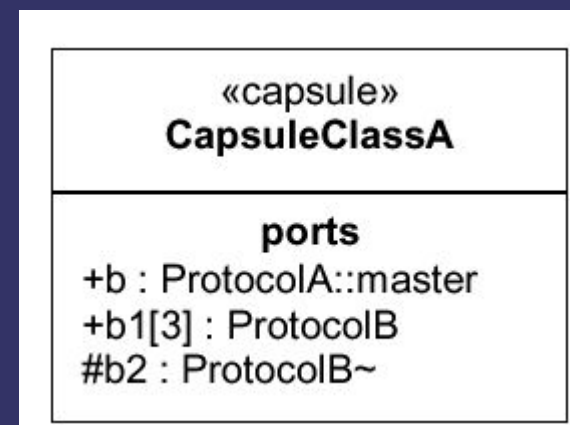
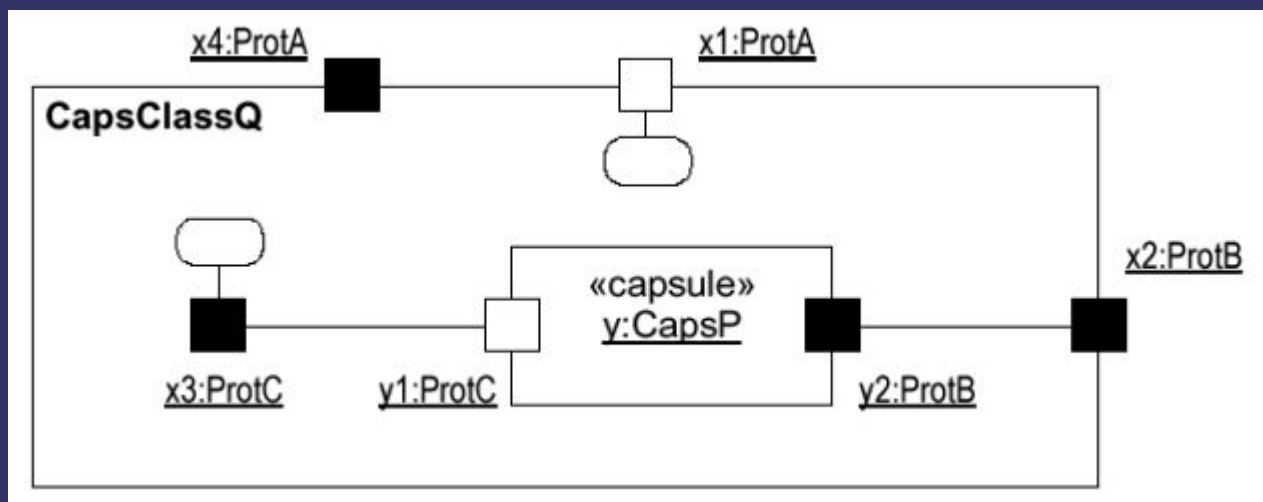
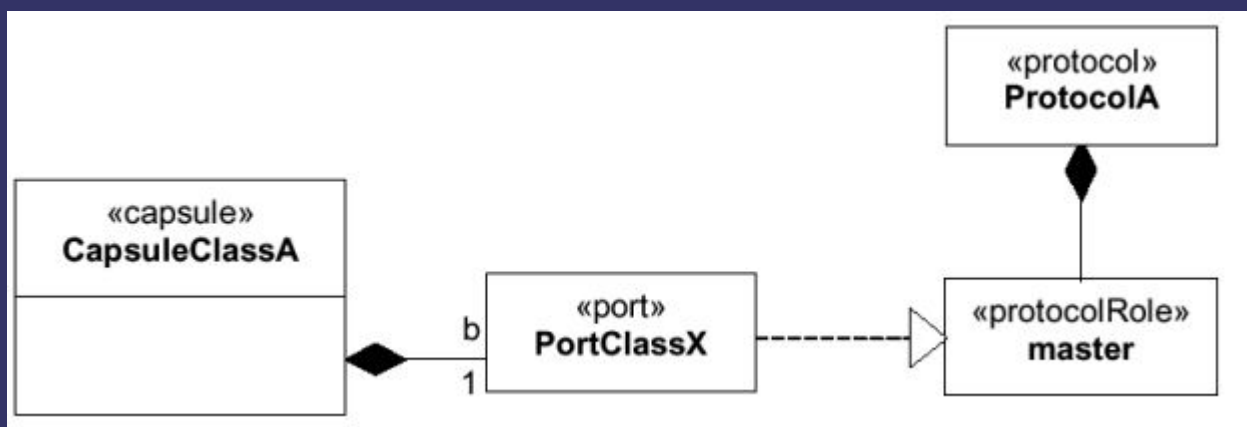
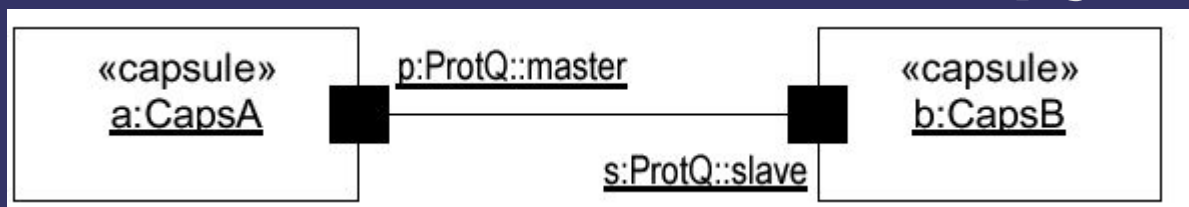


# *Диаграмма композитной структуры (Composite structure diagram)*

- статическая структурная диаграмма, демонстрирует внутреннюю структуру классов и, по возможности, взаимодействие элементов (частей) внутренней структуры класса.
- Капсула (part), черный ящик.
- Порт.
- Связь (коннектор).



# Диаграмма композитной структуры



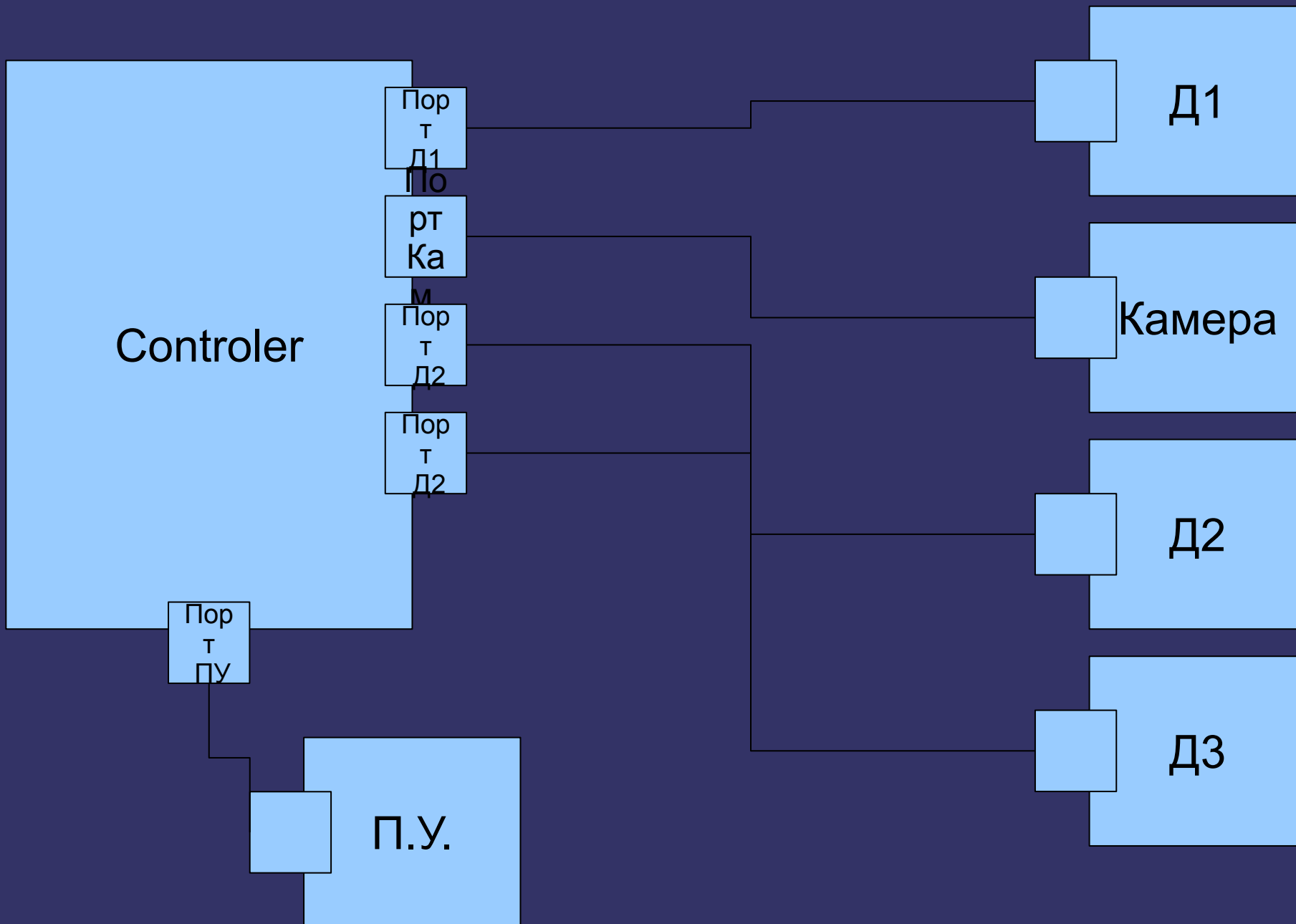
# *Пример использования Описание структуры и логики конвейера ПО.*

Дан конвейер, который имеет:

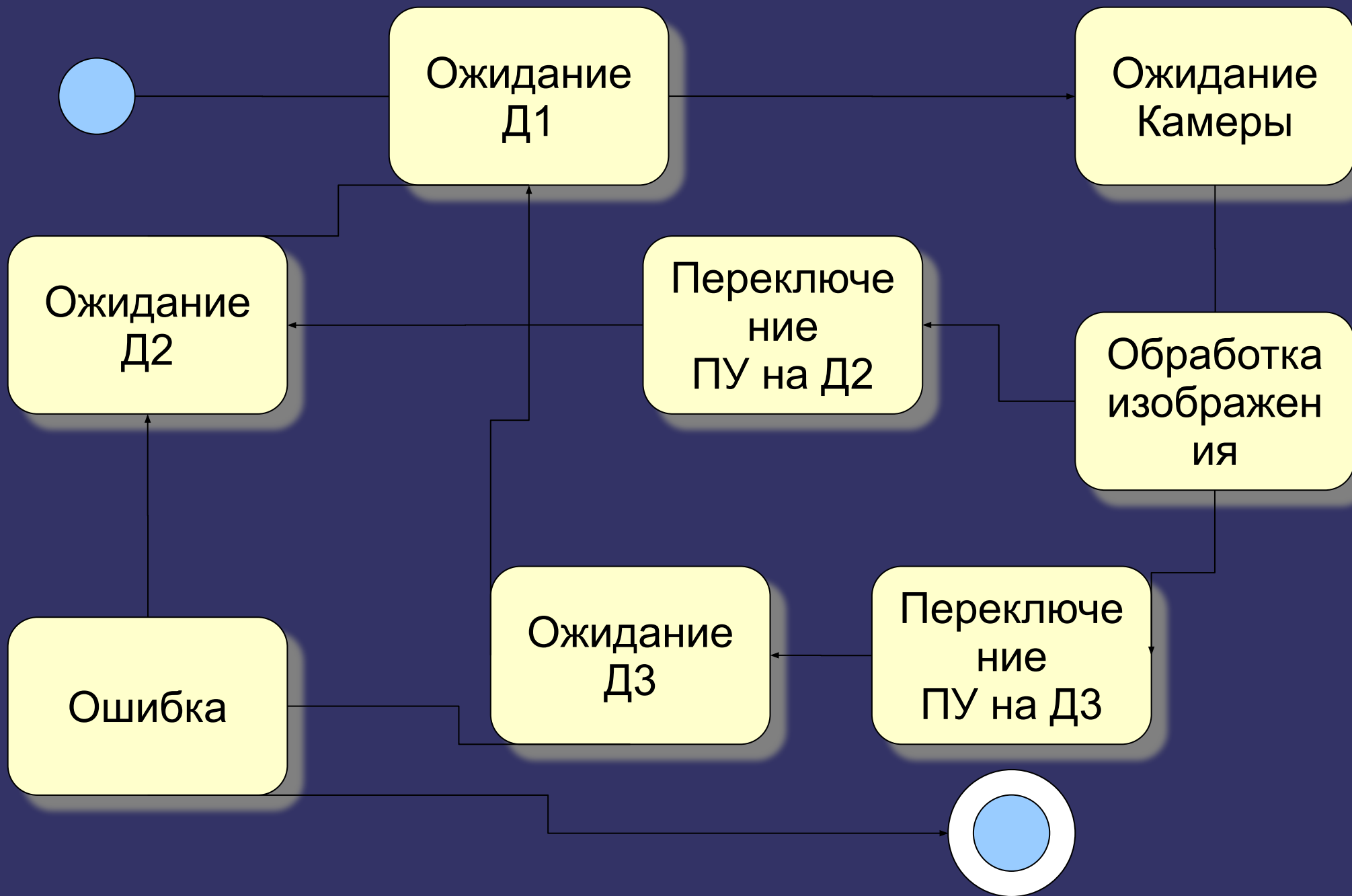
- 3 датчика наличия детали на конвейере, один входной(Д1), и два выходных (Д2, Д3).
- оптическую систему контроля качества (Камера).
- одно переключающее устройство, задачей которого является перенаправленные бракованных(сортированных) изделий на другую линию(ПУ).

Задача заключается в сортировке изделий, и защиты линии от блокирования изделий на конвейере.

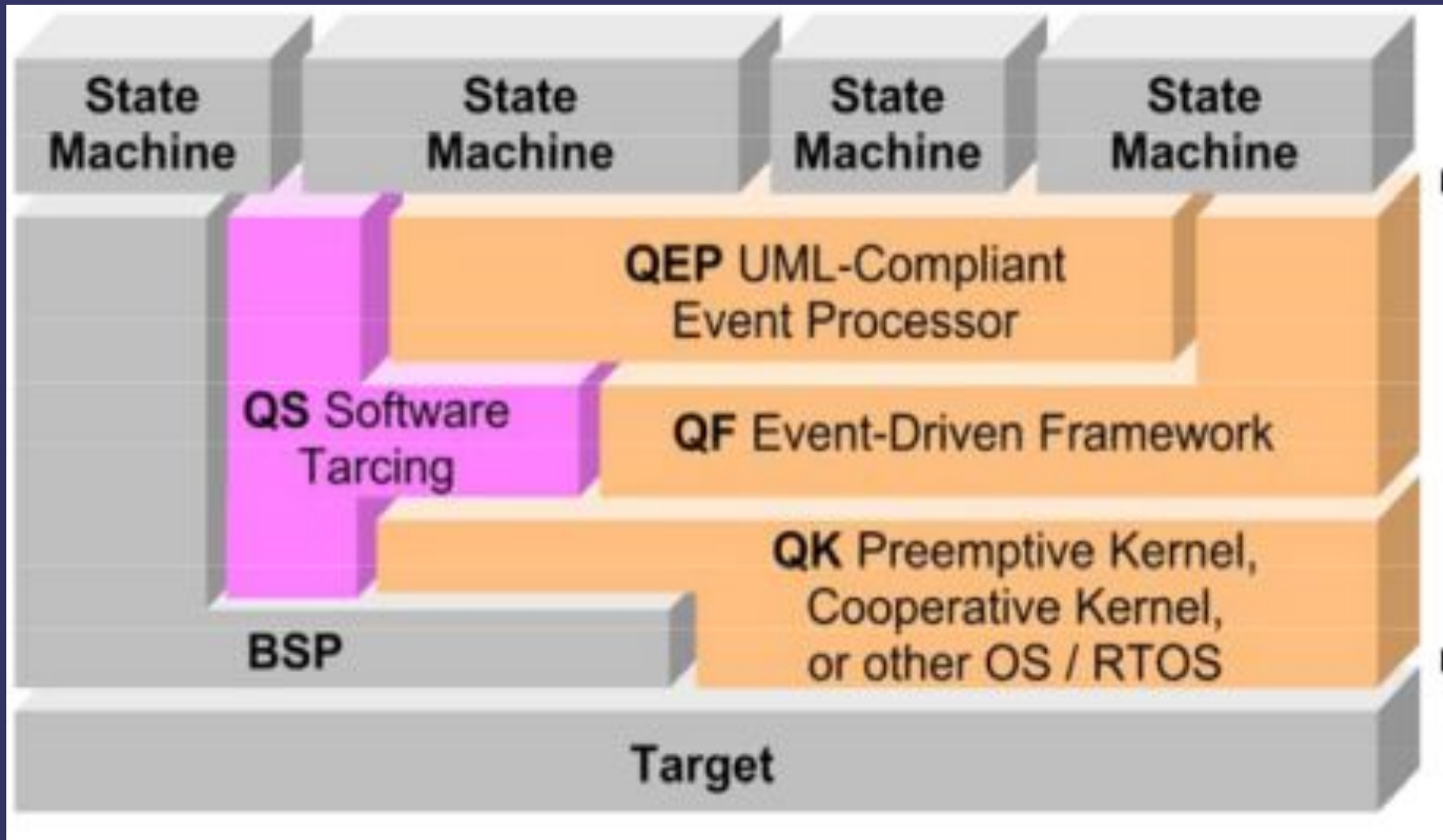
# ООМ РВ конвейера



# Диаграмма состояний конвейера



# Пример реализации QP(Quantum Processor)



# *Разъяснение обозначений*

State machine — конечный автомат,  
QEP UMP (Event Processor) — Программный  
процессор управления событиями для  
конечных автоматов.

QF Event-Driven Framework - каркас  
управления событиями

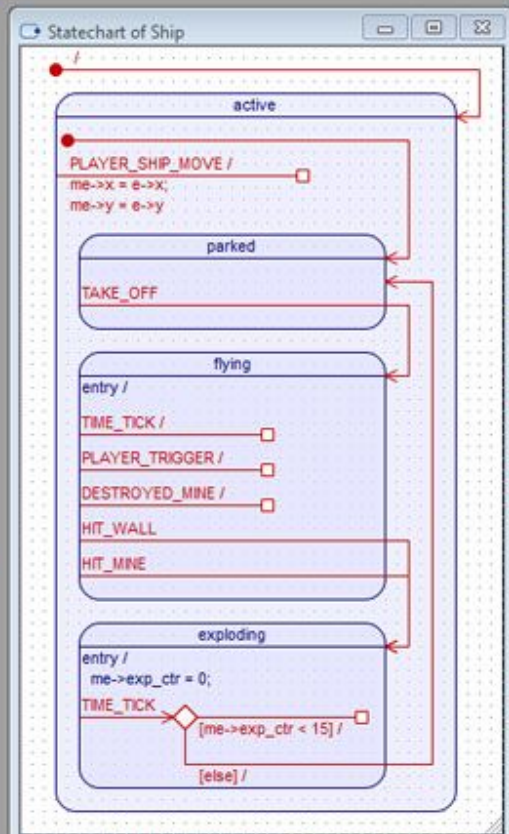
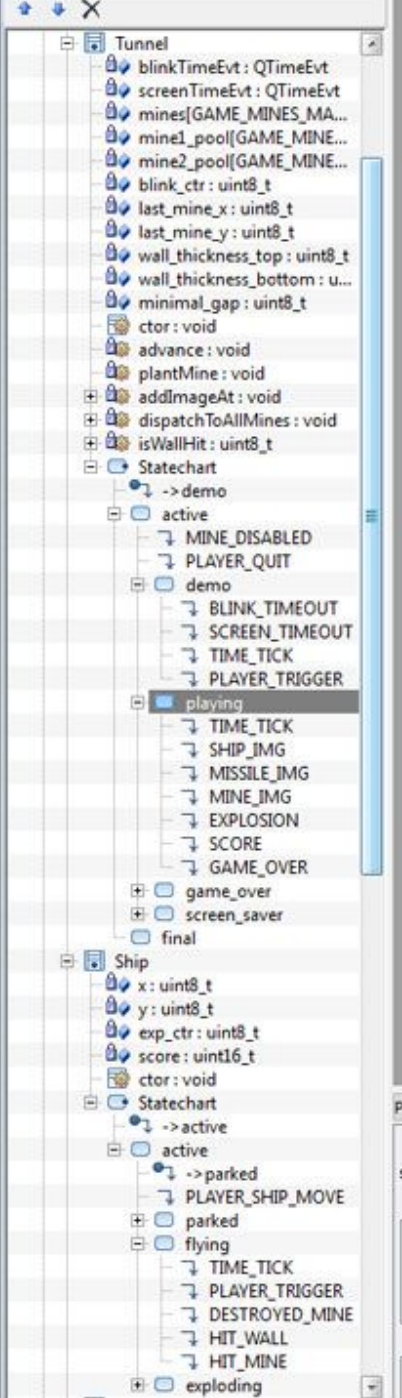
QK Preemptive Kernel ... - уровень  
распределение ресурсов.

BSP (Base System Platform) — базовая  
системная платформа

Target — аппаратное окружение или целевая  
платформа.



Explorer



```

/* local objects
static Mine1 l_mine;

#define MINE_ID(n) 0
/* Mine1 class declaration
#define(AOs::Mine1)

/* encapsulated declaration
#define(AOs::Ship)
/* local objects
static Ship l_ship; /* the ship
/* Public-scope objects ---
QActive * const AO_Ship = &l_ship;
/* Active object definition
#define(AOs::Ship)
  
```

Property Editor: State

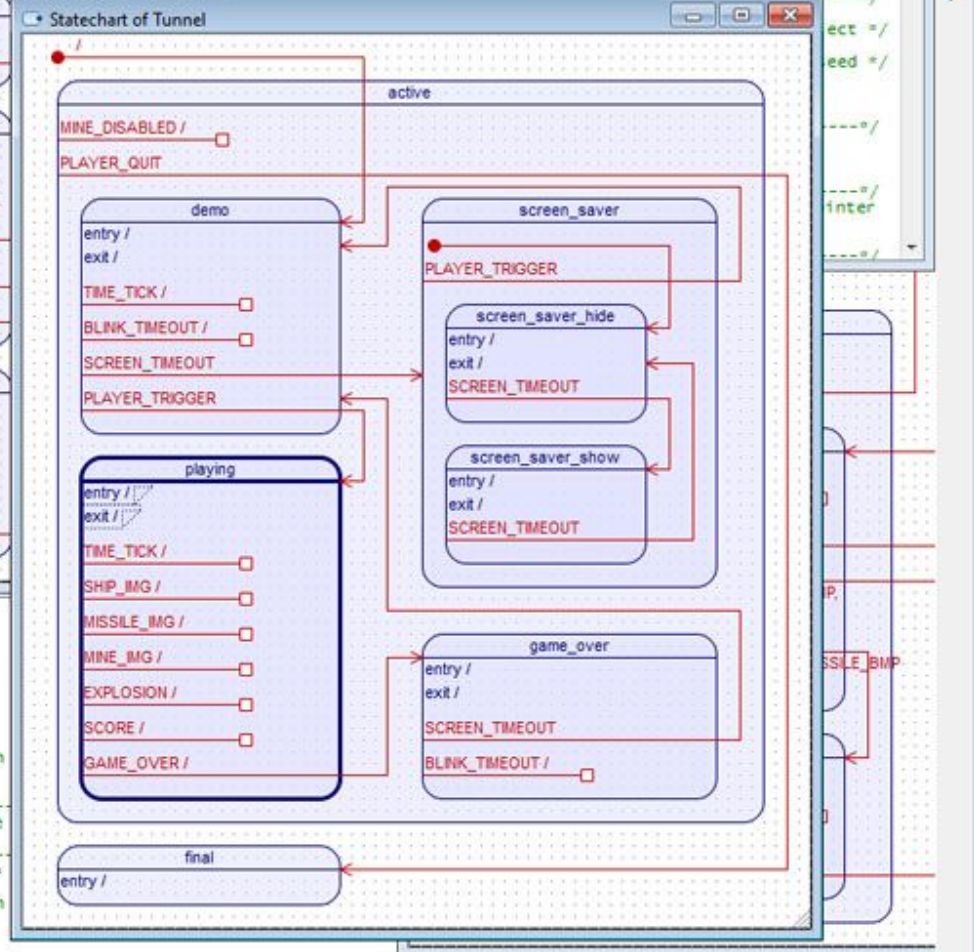
name: playing  
superstate: active

```

entry:
static QEvent const takeoff = { TAKE_OFF_SIG, 0 };
me->minimal_gap = GAME_SCREEN_HEIGHT - 3;
/* erase the walls =
memset(l_walls, (uint8_t)0,
(GAME_SCREEN_WIDTH * GAME_SCREEN_HEIGHT/8));
QActive_postFIFO(AO_Ship, &takeoff); /* post the TAKEOFF sig
exit:
QEvent recycle;
recycle.sig = MINE_RECYCLE_SIG;
  
```

```

tunnel.c
#include "qp_port.h"
#include "bsp.h"
#include "game.h"
#include <string.h>
/* for memmove() and memcpy() */
Q_DEFINE_THIS_FILE
  
```



Log

```

INFO> Code generation started (05:55:37.125 pm): entire model
INFO> (D:\software\qpc\examples\80x86\dos\watcom\l\game-qm\gm_code\ship.c): (re)generated file
INFO> Code generation ended: 1 file(s) generated, 6 file(s) processed, 0 error(s), and 0 warning(s)
  
```

*Вопросы?*



# *Ссылки:*

1. Курс «Визуальное моделирование» Лекция 6 «Визуальное моделирование систем реального времени, часть I»  
<http://www.intuit.ru/department/se/vismodtp/6/2.html>
2. Каркас для проектирования ПО в виде диаграмм состояний.  
[www.state-machine.com](http://www.state-machine.com)