

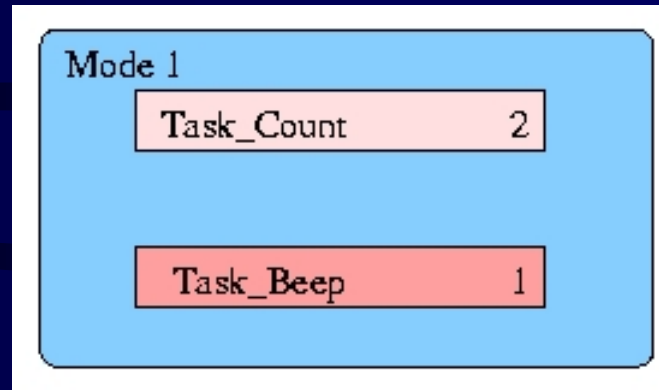
Our RTOS
(Homework #1)
EE 2900

Yang Zhao
Arkadeb Ghosal

Introduction

- ❑ A simple application
- ❑ E code and opcode
- ❑ The E interpreter
- ❑ A mode switching example
- ❑ Demos

The application



```
0: CALL count_driver;  
   CALL beep_driver;  
   SCHEDULE count;  
   SCHEDULE beep;  
   FUTURE (timer, 5);
```

```
5: CALL beep_driver;  
   SCHEDULE beep;  
   FUTURE (timer, 0);
```

enum opcode	int (*fn) ()	int arg1	int arg2
CALL	Driver		
SCHEDULE	Task		
FUTURE	Handler	Argument	Line #
COND	Driver	Line #	

```

eco[0].opcode = CALL;
eco[0].fn = &count_driver;
eco[1].opcode = CALL;
eco[1].fn = &beep_driver;
eco[2].opcode = SCHEDULE;
eco[2].fn = &count;
eco[3].opcode = SCHEDULE;
eco[3].fn = &beep;
eco[4].opcode = FUTURE;
eco[4].fn = &wait_time;
eco[4].arg1 = 1000;
eco[4].arg2 = 5;
eco[5].opcode = CALL;
eco[5].fn = &beep_driver;
eco[6].opcode = SCHEDULE;
eco[6].fn = &beep;
eco[7].opcode = FUTURE;
eco[7].fn = &wait_time;
eco[7].arg1 = 1000;
eco[7].arg2 = 0;

```

```
Emachine(em);
```

The E Interpreter

```
switch(em->eco[i].opcode) {
    case NOP:
        break;
    case CALL:
        (*(em->eco[i].fn))(0, em);
        break;
    case SCHEDULE:
        k = execi( em->eco[i].fn, 0, em, 1, DEFAULT_STACK_SIZE);
        if (k==-1) { cputs("err"); }
        break;
    case FUTURE:
        m = em->eco[i].arg1;
        (*(em->eco[i].fn))(m, 0); //returns when trigger is true
        i = em->eco[i].arg2 - 1;
        break;
    default:
        break;
}
```

An event triggered application

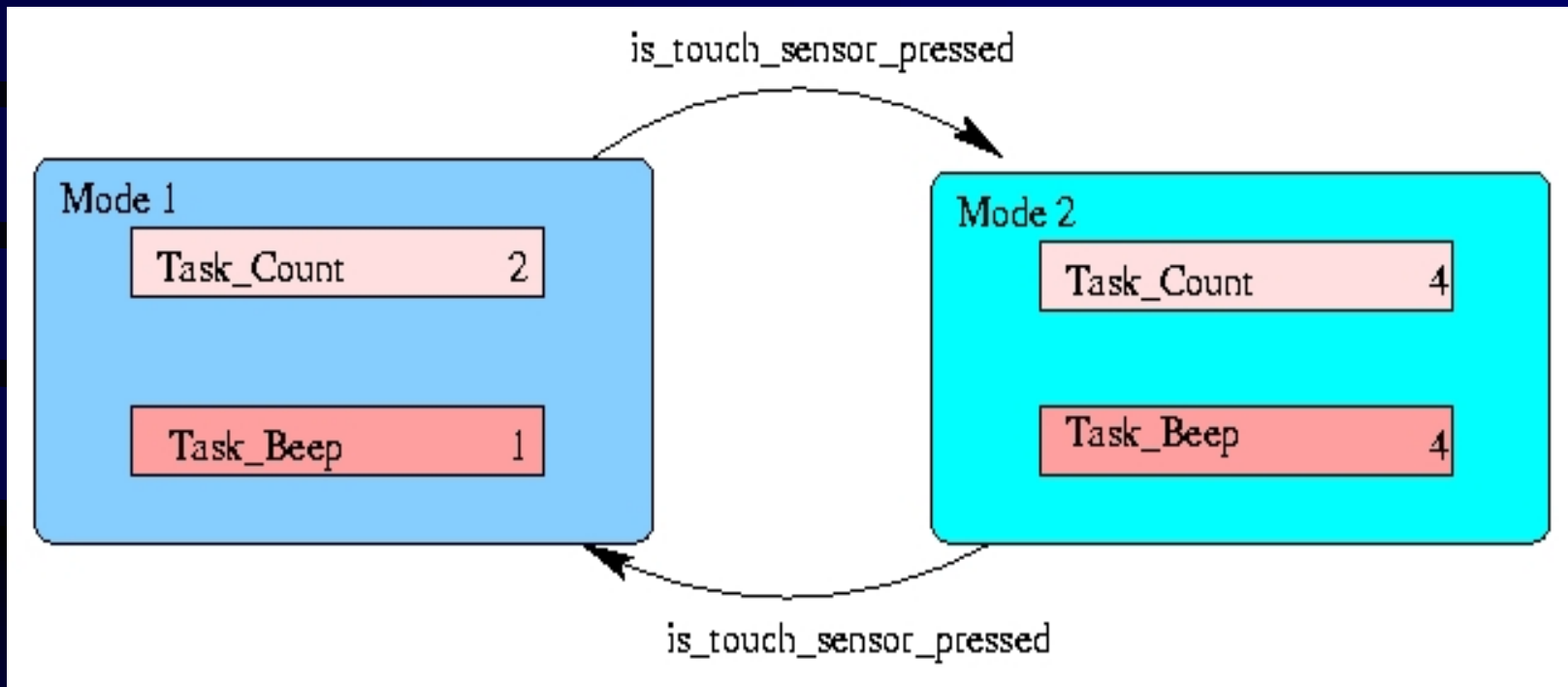
```
0: CALL count_driver;
   CALL beep_driver;
   SCHEDULE count;
   SCHEDULE beep;
   FUTURE (button, 5);

5: CALL beep_driver;
   SCHEDULE beep;
   FUTURE (timer, 0);
```

```
eco[0].opcode = CALL;
eco[0].fn = &count_driver;
eco[1].opcode = CALL;
eco[1].fn = &beep_driver;
eco[2].opcode = SCHEDULE;
eco[2].fn = &count;
eco[3].opcode = SCHEDULE;
eco[3].fn = &beep;
eco[4].opcode = FUTURE;
eco[4].fn = &wait_button_pressed;
eco[4].arg1 = BUTTON_PROGRAM;
eco[4].arg2 = 5;
eco[5].opcode = CALL;
eco[5].fn = &beep_driver;
eco[6].opcode = SCHEDULE;
eco[6].fn = &beep;
eco[7].opcode = FUTURE;
eco[7].fn = &wait_time;
eco[7].arg1 = 1000;
eco[7].arg2 = 0;

Emachine(em);
```

Mode switches !!



The E code for two modes

```
0: CALL count_driver;
   CALL beep_driver;
   CALL touch1_driver;
   COND (read_touch1, 10);
   SCHEDULE count;
   SCHEDULE beep;
   FUTURE (timer1, 7);

7: CALL beep_driver;
   SCHEDULE beep;
   FUTURE (timer1, 0);
```

```
10:CALL count_driver;
     CALL beep_driver;
     CALL touch1_driver;
     COND (read_touch1, 0);
     SCHEDULE count;
     SCHEDULE beep;
     FUTURE (timer2, 10);
```


The conditional operator

```
eco[3].opcode = COND;  
eco[3].fn = &read_touch1_driver;  
eco[3].arg1 = 10; // if toucheded
```

```
case COND:  
    m = em->eco[i].arg1;  
    if ((*em->eco[i].fn)(m, 0)) {  
        i = em->eco[i].arg1 - 1;  
    }  
    break;
```

Demo

- ❑ Simple time triggered system
- ❑ RTOS handling both time-trigger and event-trigger
- ❑ The demo illustrating mode-switches

Future work

- Modifying the mode switch mechanism
- Finding the optimum no. of arguments and E codes required
- Encoding the project problem to E code