Dual Task Code Review

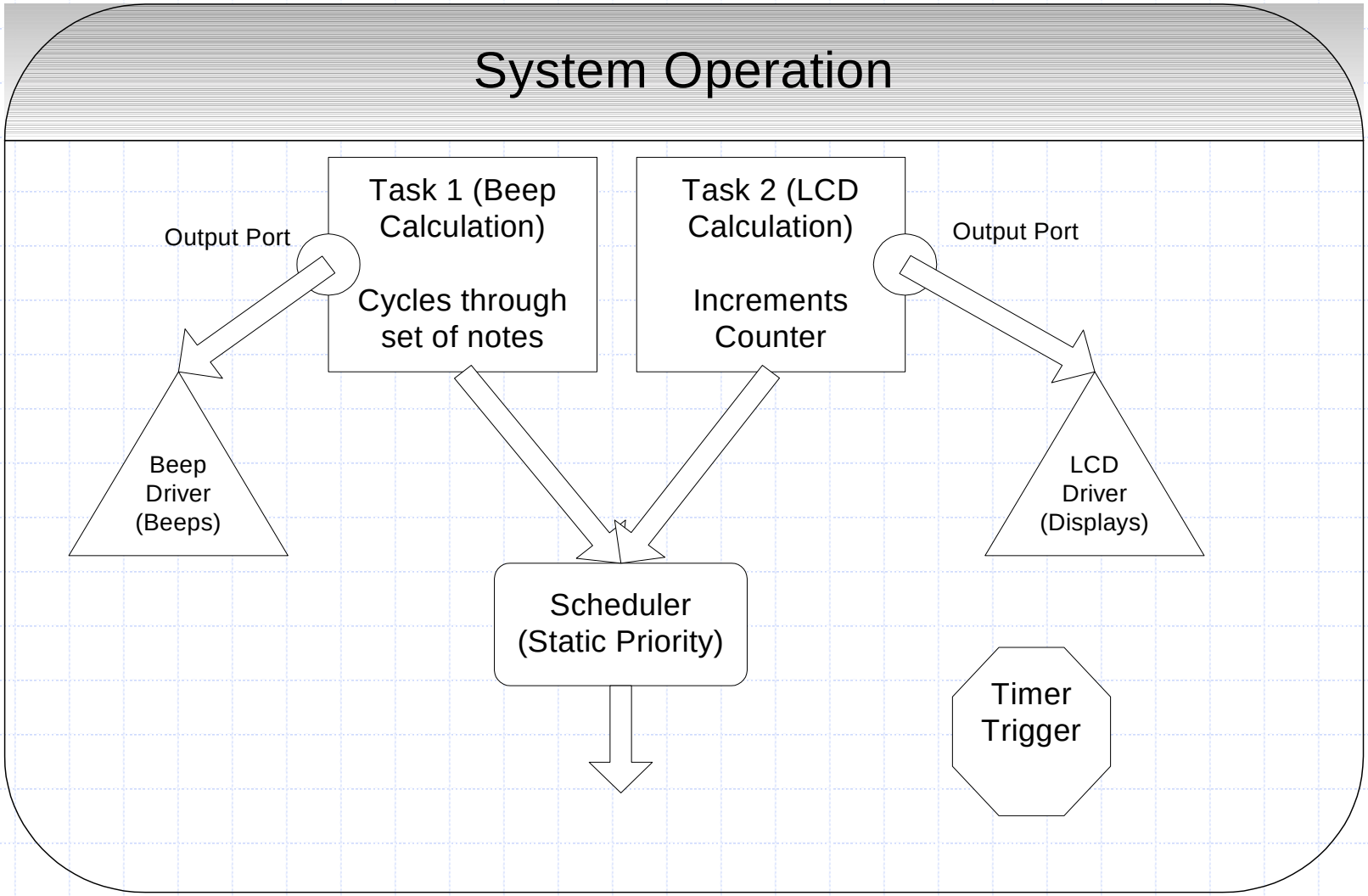
Doug Densmore

Will Plishker

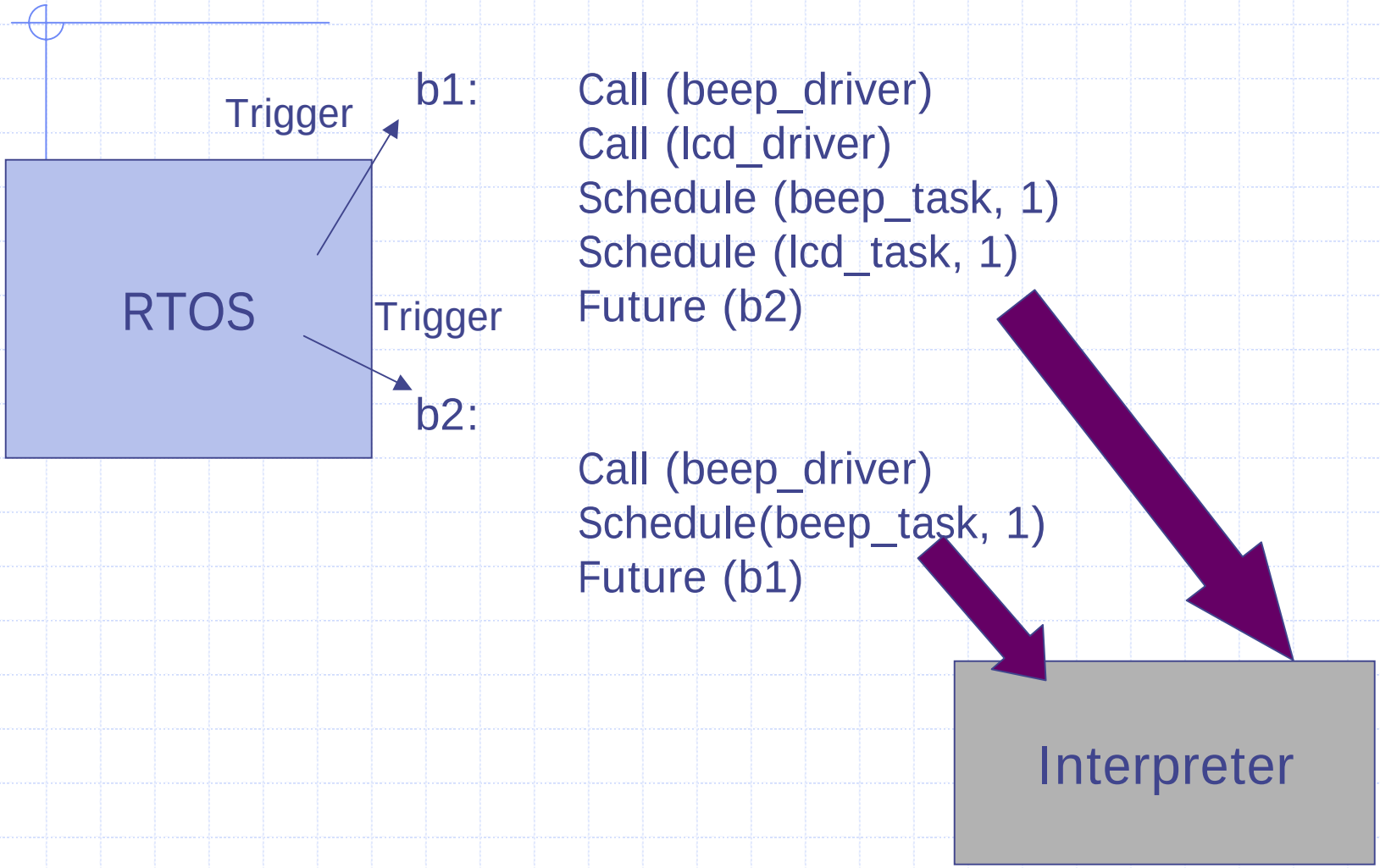
EE290o

March 19, 2002

Composition of the System



Ecode/RTOS/Scheduler



Code

```
typedef enum {
    call,
    schedule,
    future,
}Opcode;

typedef struct {
    void(*(fp))();
    int priority;
}Task;

typedef struct {
    Opcode opcode;
    int index;
    void(*driver)();
    Task task;
}Ecode;
```

```
int main(int argc, char **argv) {
    while(1) {
        if(rtos_timer>35) {
            rtos_timer= 0;
            while(ecode[i].opcode!=future) {
                if(ecode[i].opcode==call)
                    (*(ecode[i].driver))();
                if(ecode[i].opcode==schedule){
                    scheduled_tasks[head_scheduled_tasks] = ecode[i].task;
                    head_scheduled_tasks++;
                }
                i++;
            }
            i = ecode[i].index;
            execute();}
    }
    return 0;
}
void execute() {
    while(head_scheduled_tasks) {
        head_scheduled_tasks--;
        scheduled_tasks[head_scheduled_tasks].fp();
    }
}
```