

---

# **CS 460**

## **PARKING STRUCTURE**

### **MANAGEMENT OF PARKING SPOTS**

PROJECT #81

28 AUGUST 2021



## 2 PARKING STRUCTURE EXAMPLES

---

- At the Sunport
- New at UNMH
- Yale structure?

### 3 PROBLEM DEFINITION

---

- Functionality is to be determined through discussions with the teaching team acting as UNM hiring your company
- The main deliverable is a faithful functional simulation using the control software
- Appropriate software engineering processes must be pursued during development

## 4 PROJECT SCHEDULE

---

- Each milestone is to be completed in one week
- Deliverables
  - Requirements Definition Document (RDD)
  - Software Requirements Specification (SRS)
  - Software Architecture Design (SAD)
  - Code in the GitHub repository
- Demonstration (week 7)



## 5 SPECIAL PEDAGOGICAL NOTE

---

- You are engaging in a project without having completed the prerequisite training
- Technical assistance will be provided at every step
- This is not atypical for what happens in an industrial setting—on the job training
- The pedagogical goal is
  - to expose you to the complexities of software development by using what appears to be a simple problem
  - to help you understand that software is an integral part of the system it controls



## SAMPLE TABLE OF CONTENTS FOR EACH REPORT

---

# 7 REQUIREMENTS DEFINITION DOCUMENT (RDD)

---

1. Introduction
2. Objectives
3. System Organization
4. Capabilities
5. Design Constraints



# 8 REQUIREMENTS DEFINITION DOCUMENT (RDD) (1)

---

## 1. Introduction

- purpose of this report is to describe the control system that your company offers in preparation for formulating requirements for software
- high level overview of capabilities
- report organization (last paragraph and refer to section numbers)



## 9 TABLE OF CONTENT FOR RDD (2)

---

### 2. Objectives

- introductory paragraph
- bulleted list of objectives for developing this control system,  
e.g.,
  - Have it be a showcase for other parking structures...
  - use this as a steppingstone to enter the market



# 10 REQUIREMENTS DEFINITION DOCUMENT (RDD) (3)

---

## 3. System Organization

- introductory paragraph
- pictures of the parking structure, entrance and a typical floor with the parking spots
- identify all components that the software will need to interface using a diagram consistent with the picture, but which may refer to specific components (e.g. sensors, entrance screen, lights, etc.)

# 11 REQUIREMENTS DEFINITION DOCUMENT (RDD) (4)

---

## 4. Capabilities

- introductory paragraph
- bulleted list of features provided by the system (logically organized)
- **do provide functionality details but at the right level of abstraction (what it will do, not how)**
- consistency with the objectives (what will help enter the market? E.g.: what kind of sensors to use, which are the most reliable to detect a car, safety concerns, will you track the cars as they are driving?)

## 12 REQUIREMENTS DEFINITION DOCUMENT (RDD)

---

### 5. Design Constraints

- anything relevant regarding the control software, e.g., Java



## 13 ISSUES TO PONDER

---

- What do I need to do when I install the system in my parking structure? Where will it be hosted?
- Will be register the license plate?
- Will it be designed for a parking with payment?
- What happens when there is no power?
- What happens if the light is green but there is a parked motorcycle? Will the sensors detect a motorcycle?



## TEAM FORMATION

---



## 15 FORMING THE TEAMS

---

- 51 students currently registered
- 11 teams (7 teams w/ 5 members + 4 w/ 4 members)
- members of each team + one of the members is the manager of the team.
- The teams will be posted on canvas and groups will be created, all the assignments are evaluated by team, except for participation in lectures and workshops which is individually evaluated.



# 16 SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

---

## 1. Introduction

- purpose of this report is to formalize the requirements for the control software
- high level overview of the role the software plays
- report organization (refer to section numbers)

## 2. General Description

- block diagram showing all the physical interfaces

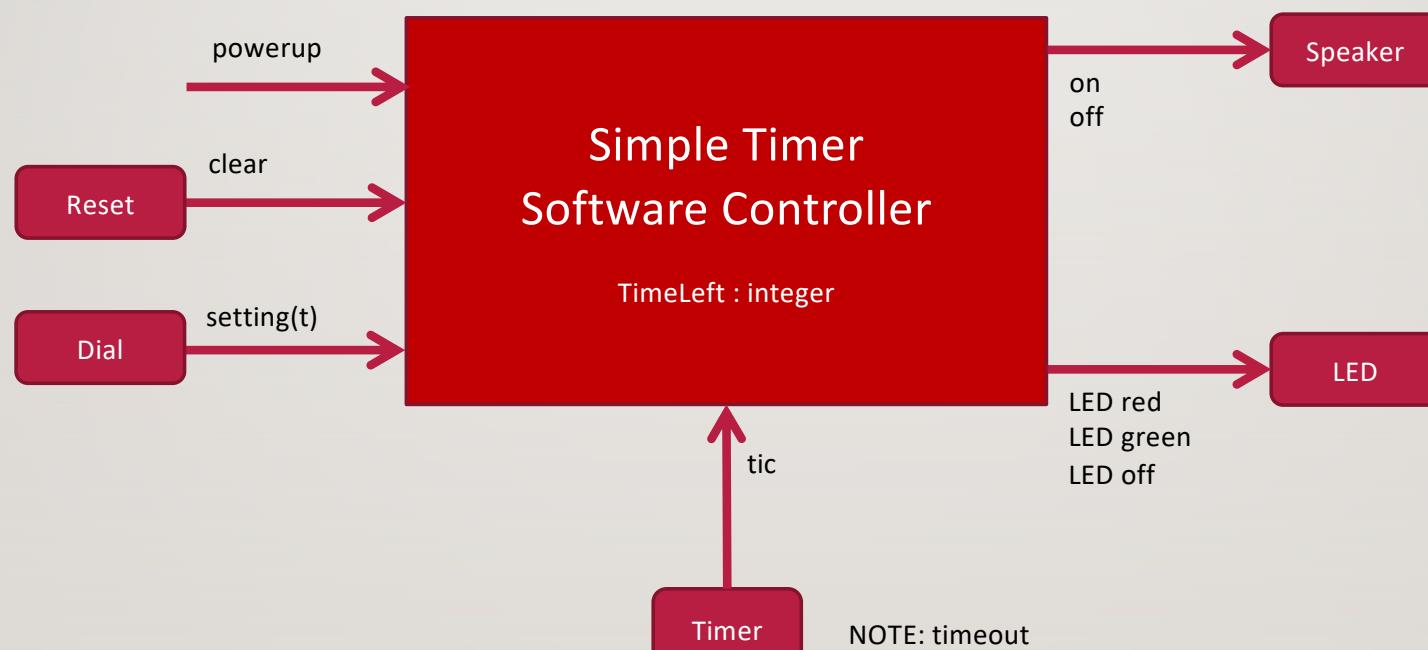
## 3. Specific Requirements

- diagram showing all the logical interfaces to the software consistent with the block diagram
- explanation of all the logical interfaces
- finite state diagrams capturing formally the behavior of the system

## 4. Design Constraints

- relevant constraints (e.g., object-oriented design, Java)

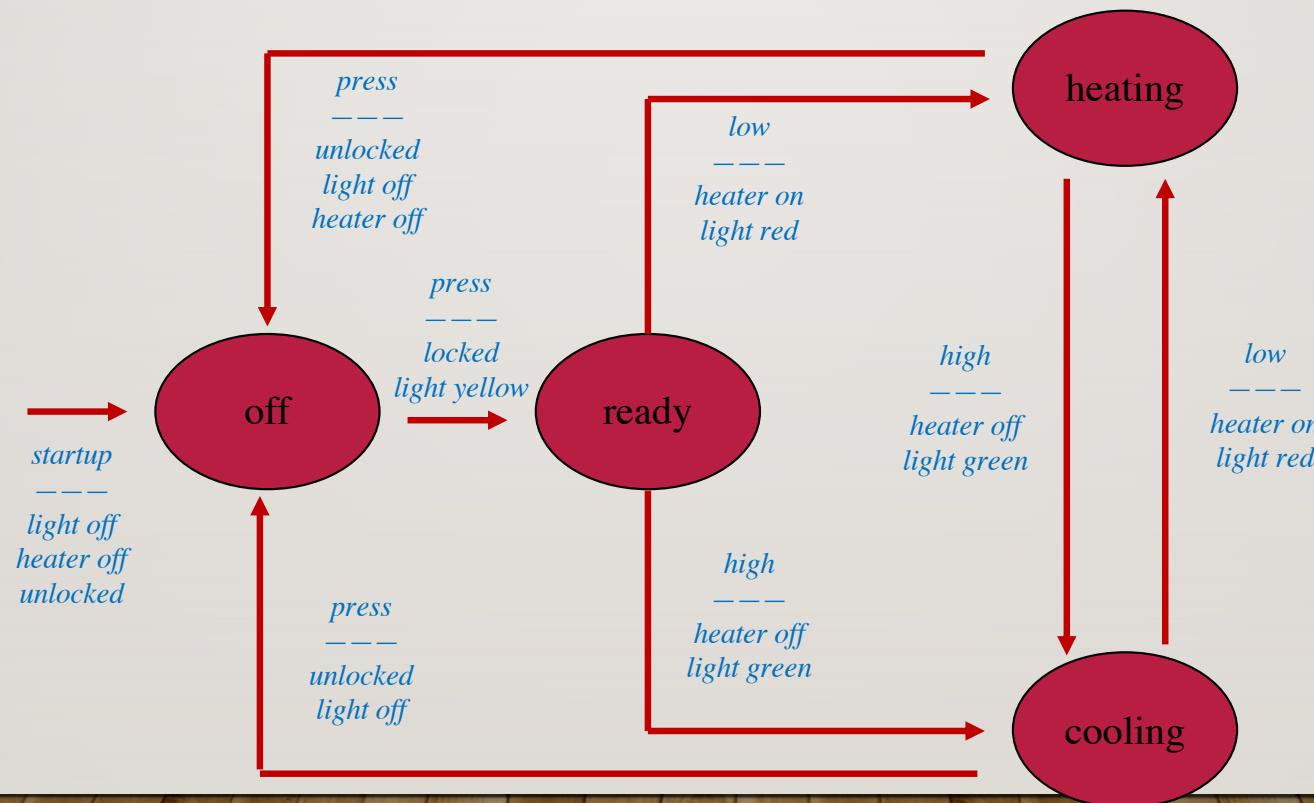
# LOGICAL DIAGRAM



# SAMPLE FINITE STATE MACHINE

HVAC control logic

We use the OMT notation available on canvas



# 19 SOFTWARE ARCHITECTURE DESIGN (SAD)

---

- 1. Introduction
- 2. Design Overview
  - design diagram and explanations
- 3. Component Specifications
- 4. A Sample Use Case



# DESIGN NOTATION

---

