# Concordia University
# Department of Computer Science and Software Engineering

**SOEN 422 Fall 2021**

—

**Project Report Guideline**

November 18, 2021

**Abstract**

This project report guideline presents a simplified template following the organization and the format of the IEEE Standard 1016-2009 and recommended practice for Software Design Descriptions (SDDs) [1].

## 1 Project Report Template

The Project Report is composed of five sections:

1. **Title Page:**

   The title page should include the team name, team members and students id numbers.

2. **Introduction: (1/2 page)**

   This section provides a short description of your project. It should define the scope and purpose of project. Discuss what hardware and software were used.

   The purpose statement should summarize the specific topic and goals of the project. It should be:

   - Detailed and precise: not general, broad or obscure
   - Concise: one or two sentences
   - Clear: not vague, ambiguous or confusing
   - Goal-oriented: stated in terms of desired outcomes

   The scope statement should:

   - Define the boundaries of the project.
   - Define the expected outcome of the project.
   - Identify hardware and software constraints for developing the project.

3. **Goal: (1/2 page)**

   This section provides a quick overall summary (goal of the project) of what your group's tasks have achieved during the port of the virtual machine on the Arduino Nano.

4. **Body of the Report: (5 pages)**

    - **Context Design Viewpoint: Key Concept Model of the Project: (1 page)**
      This section should summarize and explain the final version of your key concept model of the project. An example is given in Section 2. The block must have between 8 up to 15 blocks (half page) and explanations (half page).
    - **Composite Design Viewpoint: Package Diagrams of the Project: (2 pages)**
      This section should summarize and explain your package diagrams with all their dependencies in the project. Examples are given in Section 2. All C or C++ software files used in the project must be represented. All package diagrams must be represented on one page (as an overview) and all the explanations on the second page.
    - **Implementation of Tasks: (5 pages)**
      This section should summarize your implementation of the main (five) tasks. Theses (tasks) subsections are expected to have the discussion of the implementations as well. You should include snippets of code that your group feels are relevant (and proud) to your project's overall functionality. Do not simply toss in code but rather, discuss what is relevant about the code of the specific task and where it belongs. Theses subsections serve also as a reflection of the tasks. Discuss which tasks were met, which weren't and why.

5. **Conclusion: (1/4 page)**
   The conclusion should tie everything together. Filling your conclusion with things such as "The project was fun and entertaining" does not add value. Make your conclusion meaningful.

**Your submission is expected to be in a *.zip* file on Moodle (no link to a repo). Within this archive, your final report and all your code is to be submitted neatly in a folder. The report is to be in a *.PDF* file.**

Naming convention for zip file is: `ProjectReport_TeamXX.zip`

# 2 Design Viewpoint Examples

.
The following section presents examples of design viewpoint which complements the *Lecture 9* notes.

## 2.1 Context Design Viewpoint: Key Concept Model

### Key Concept Model

The key concept model is a graphic representation in the form of a block diagram, which is a simplified class diagram. This diagram is composed of rectangles representing the system's fundamental and essential concepts (objects) and the links between these concepts. A link is a simple straight line involving coupling between two concepts. The resulting diagram of the key concept model is a first draft that will eventually evolve (after several iterations) in class or package diagram(s).

**Example:** A pedestrian crossing signals application on an MCU.

A pedestrian crossing signals application uses a bargraph component with 3-LEDs. There is a button to cross the street. When you press the button, it turns on the button LED and then starts the ”Walk” effect on the bargraph to display a rotating effect, letting you knowthat pedestrianscan cross the street.The walk effect will change to the ”don’t walk” or ”wait” effect after 10 seconds. The don’t walk/wait effect flashes the 3-LEDs rapidly for 5 seconds and turns all LEDs off.

The following is the corresponding key concept model as a first draft:

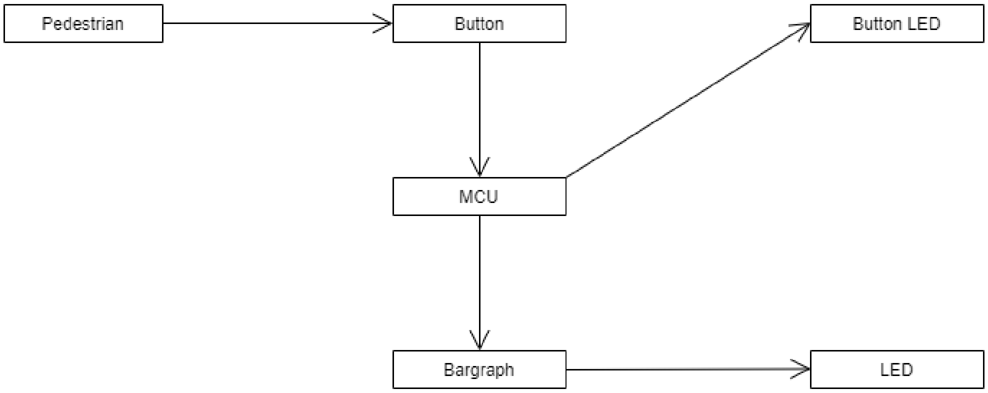


Figure 1: Key Concept Model (first draft) of the Pedestrian Crossing Signals Application

The following is the corresponding key concept model as the final version:

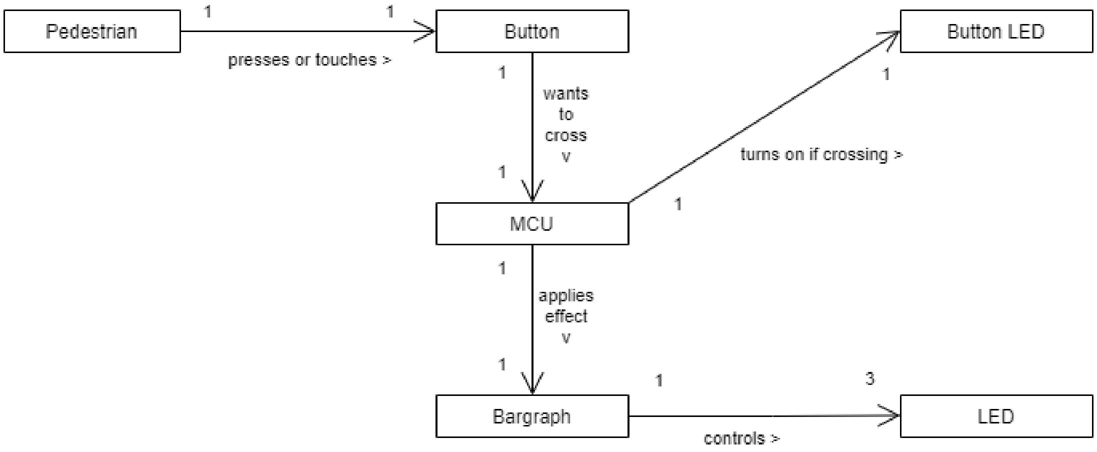
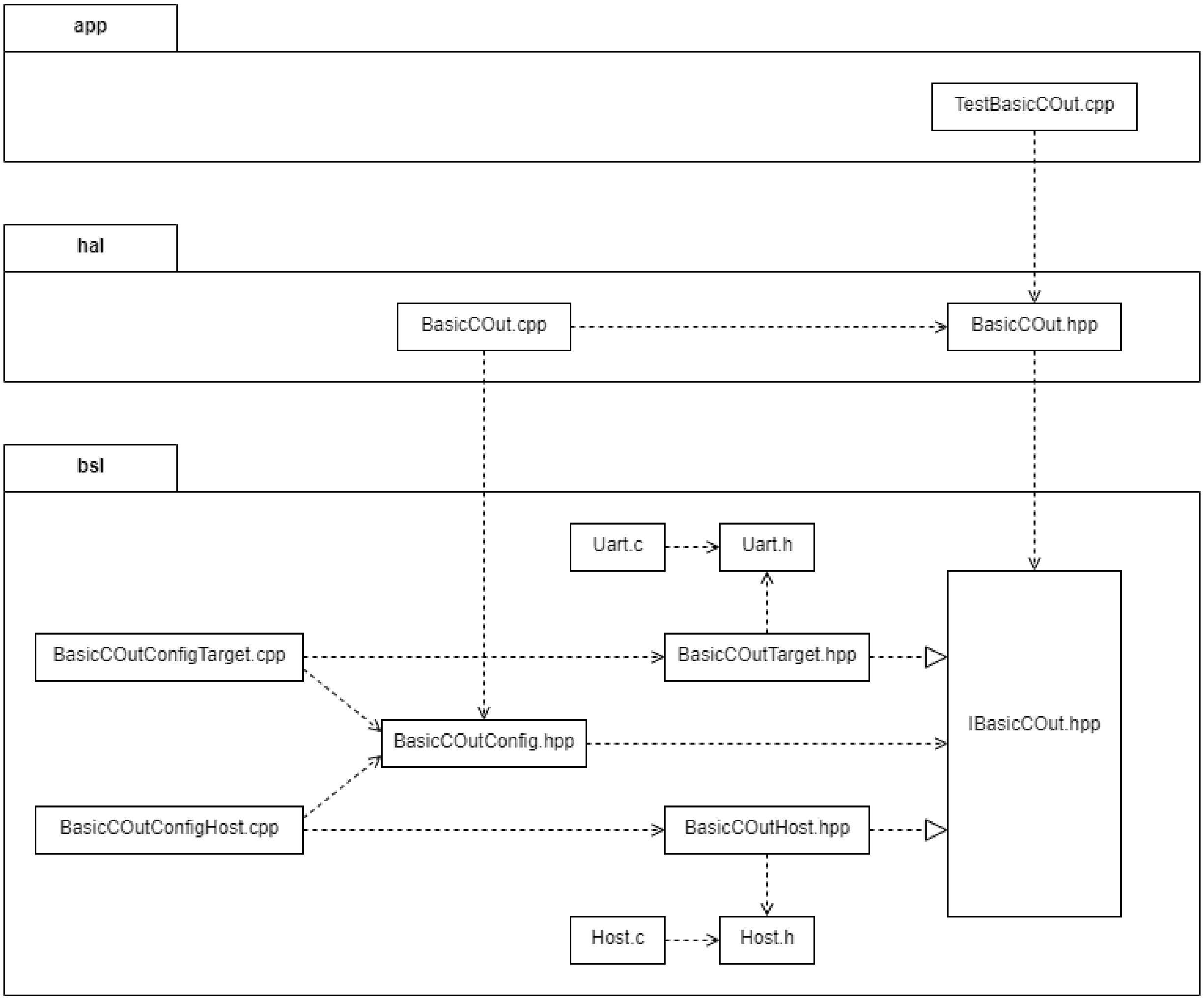


Figure 2: Key Concept Model (final version) of the Pedestrian Crossing Signals Application

## 2.2 Composite Design Viewpoint: Package Diagrams and Dependencies

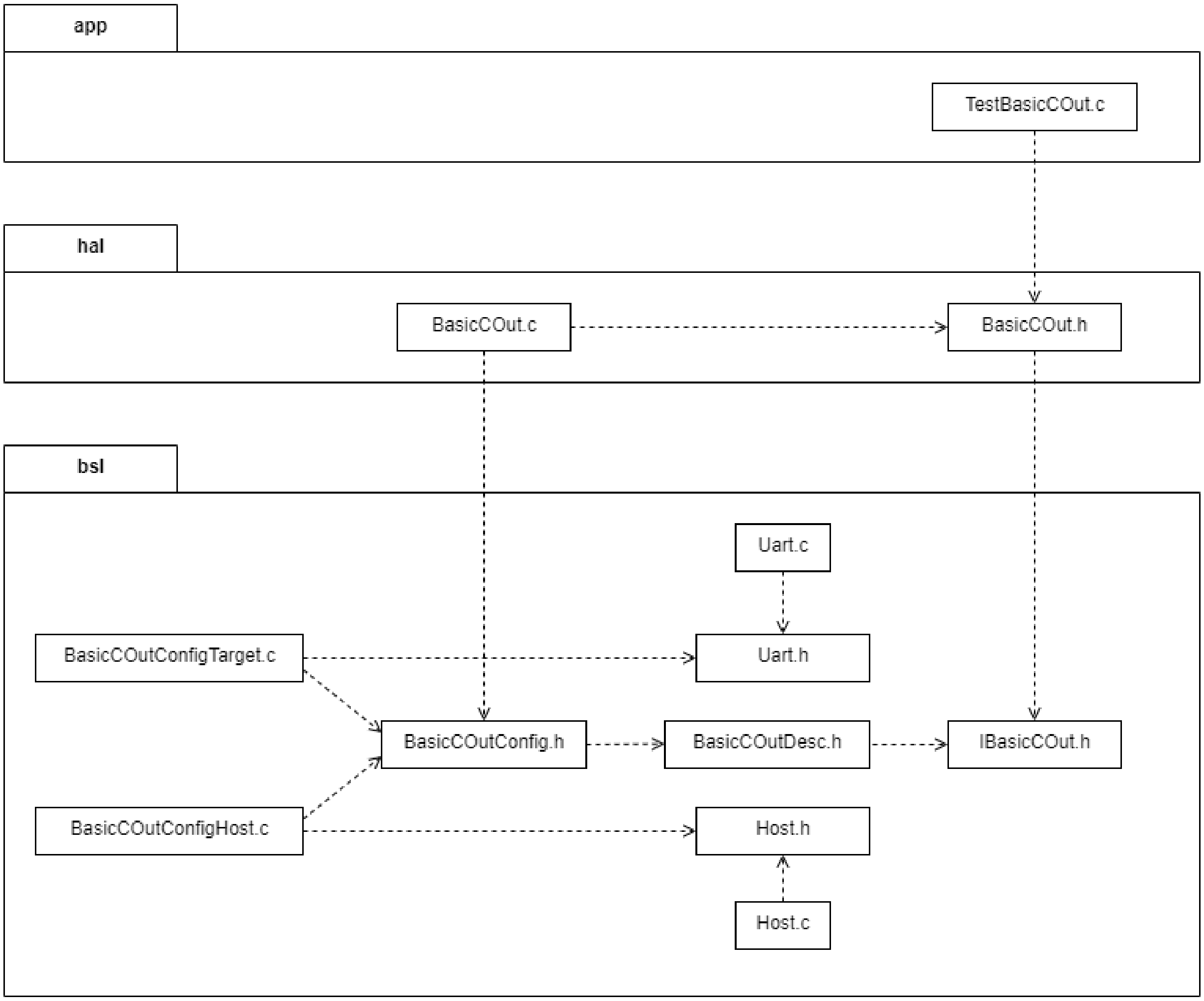
The following is an example of package diagrams with all their source files' dependencies in C++[2]:

### Dependencies Between Layers in C++



### Dependencies Between Layers in C

The following is an example of package diagrams with all their source files' dependencies in C [2]:



## 3 References

[1] IEEE Std 1016-2009, IEEE Standard and Recommended Practice for Software Design Descriptions. 2009/07/20.

[2] Michel de Champlain, Dependencies Between Layers for a Basic Console Output Component in C/C++, SOEN 422 --- Embedded Systems and Software, Fall 2021, November 17, 2021.