

## TASK1: LED Counter Learning

### Objectives:

- Become familiar with PLPTool features
- Use PLP arithmetic and control flow instructions
- Write to a memory mapped I/O device

### Installing a Java Runtime Environment:

In order to install PLPTool you will need to have a JRE (Java Runtime Environment) installed. The default installation of a JDK (Java Development Kit) includes a JRE if you have one installed then you are ready to install PLPTool. If you don't have a JRE installed you can use one of the following links to get it:

1. [Download JDK](#) – I would recommend installing the JDK if you haven't already since it is required in order to compile Java programs (you won't need to write any Java in this class, but other classes will most likely use it).
2. [Download JRE](#) – Use this instead if you don't need or don't have space for the entire JDK.

### Installing PLPTool:

There are 4 PLPTool releases available on GitHub. The different releases are as follows:

- **Windows (.exe)** – Uses a standard installation wizard and can associate *.plp* files with PLPTool
- **Debian (.deb)** – Install using a debian package manager
- **Universal (.zip for 32-bit and 64-bit)** – This is the version that can be used on **macOS** and **non-Debian Linux** distributions. The zipped folder contains PLPTool as an executable JAR file. It also contains additional files and directories such as example PLP projects, PLP libraries, extensions, and scripts. This version does not associate *.plp* files with PLPTool, but if you are interested in a workaround for file association on macOS, please ask the instructor.
  - Mac and Windows – Run by double clicking on JAR file, *PLPToolStatic.jar*.
  - Linux – In the terminal use the following command in the terminal within the directory of the JAR file:
    - `java -jar PLPToolStatic.jar`

All versions of PLPTool described above can be downloaded from GitHub [here](#).

**NOTE:** When you run PLPTool there will likely be a warning in the terminal or console regarding the **RXTX version**. You can disregard this warning because we will not be utilizing the RXTX library in this class.

### Creating a New PLP Project:

Once you have installed an opened PLPTool you can create a new project by clicking the *New Project* button as shown in the following screenshot.



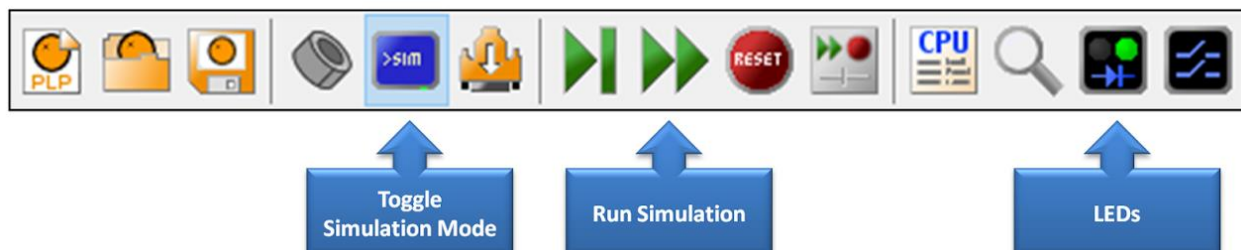
### The Task:

Write a program in PLP assembly that counts up by one starting from zero (or one) inside a loop and writes this value to the LEDs every time the value is increased. The memory address of the LEDs is 0xf0200000. The table below shows the meaning and an example usage of the instructions covered in the video, *PLP Instructions for Project 1*.

Instruction	Example Usage	Meaning
Load Immediate	<code>li \$t0, 8</code>	Register \$t0 is set to the value, 8.
Store Word	<code>sw \$t2, 0(\$t1)</code>	The value in register \$t1 is used as the memory address. The value in register \$t2 is copied into this memory address.
Add	<code>addiu \$t4, \$t3, 29</code>	Register \$t4 is assigned the sum of 29 and the value in register \$t3.
Jump	<code>j your_label_name</code>	The program jumps to the line following the label, "your_label_name:".
Label	<code>your_label_name:</code>	Defines a label called "your_label_name:" that can be jumped to.

The image below shows the buttons that will allow you to enter simulation mode, view the LEDs, and run your program. Two things to keep in mind regarding simulation mode:

1. Simulation controls will, such as run, are only visible when simulation mode has been toggled on.
2. The program cannot be edited while simulation mode is toggled on.



**Additional Resources:**

- [PLP Online Manual](#) including the following sections:
  - [Instruction Set](#) – Information about the purpose and functionality of PLP instructions
  - [Implementation](#) – Information about the I/O devices that can be simulated with PLPTool

**Deliverables:**

1. Take the Syllabus Quiz (1 point)
2. Take the Project 1 Pre Quiz (1 point)
3. Submit your program on Canvas with the format: *Firstname\_Lastname\_project1.plp* (8 points)