Building LCM from Source (Windows)

(Tested on clean Windows 10 Home 64-bit virtual machine)

Install prerequisites

1. JAVA

- Install jdk-8u191-windows-x64.exe from https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
- Set JAVA_HOME environment variable to C:\Program Files\Java\jdk1.8.0_191
- Add C:\Program Files\Java\jdk1.8.0_191\jre\bin to the PATH environment variable
- Add C:\Program Files\Java\jdk1.8.0_191\jre\bin\server to the PATH environment variable
- Add C:\Program Files\Java\jdk1.8.0_191\bin to the PATH environment variable

2. Python

- Install Anaconda Python from https://repo.anaconda.com/archive/Anaconda3-5.3.0-Windows-x86_64.exe
- Add path to Anaconda Python executable to PATH environment variable (default is %USERPROFILE%\Anaconda3)
- Open Anaconda Prompt
    - Upgrade pip package manager by running “python -m pip install --upgrade pip”
    - Install pyjnius package by running “pip install pyjnius”

3. LCM

- GLib
    - Install msys2-x86_64-20180531.exe from https://www.msys2.org/
    - Run MSYS2 by running C:\msys64\msys2_shell.cmd
    - In the MSYS window, run “pacman -S mingw-w64-x86_64-gtk3”
    - Add C:\msys64\mingw64\lib to PATH environment variable
    - Add C:\msys64\mingw64\bin to PATH environment variable
- CMake:
    - Install cmake-3.13.0-rc2-win64-x64.msi from https://cmake.org/download/
- Visual Studio 2017
    - Install Visual Studio 2017 Community from https://visualstudio.microsoft.com/downloads/
        - Install “Desktop Development with C++”
- Download and unzip lcm-1.4.0 from https://github.com/lcm-proj/lcm/releases

- Fix FindGLib2 script (does not work on Windows as released):
  In …\lcm-1.4.0\cmake\FindGLib2.cmake, line 26, replace

```
find_library(GLIB2_${VAR}_LIBRARY NAMES ${LIB}-2.0 ${LIB})
```

  with

```
if(WIN32)
  set(CMAKE_FIND_LIBRARY_SUFFIXES ".dll.a")
  set(CMAKE_FIND_LIBRARY_PREFIXES "lib")
endif()
find_library(GLIB2_${VAR}_LIBRARY NAMES ${LIB}-2.0 ${LIB})
```

- Fix LCM source code (does not work on Windows as released)
  In …\lcm-1.4.0\lcmgen\emit_go.c, replace

```
#include <unistd.h>
#ifdef WIN32
#define __STDC_FORMAT_MACROS  // Enable integer types
#endif
```

  with

```
#ifndef WIN32
#include <unistd.h>
#endif
#ifdef WIN32
#define F_OK 0                /* Test for existence.  */
#define __STDC_FORMAT_MACROS  // Enable integer types
#endif
```

  Also, in …\lcm-1.4.0\windows\WinPorting.cpp, include winsock2.h **before** Mswsock.h, i.e., replace

```
#include <Mswsock.h>
#include <stdio.h>
#include <winsock2.h>
```

  with

```
#include <stdio.h>
#include <winsock2.h>
#include <Mswsock.h>
```

- Create project using CMake
  - Open CMake (cmake-gui)
  - Point “Where is the source code” to the …\lcm-1.4.0
  - Point “Where to build the binaries” to …\lcm-1.4.0\build (it should not yet exist)
  - Click “Configure” and click “Yes” to create the build directory if it doesn’t exist
  - Specify “Visual Studio 15 2017 Win64” as the generator and use default native compilers
  - Once “Configuring Done” appears…
    - Check the “Advanced” box in the upper section of the CMake GUI
    - Scroll down in the list of CMake options and uncheck “LCM_ENABLE_TESTS”
    - Click “Configure” again
  - Click “Generate”

- Build lcm in Visual Studio 2017
    - Open …\lcm-1.4.0\build\lcm.sln (can be done by clicking “Open Project” in CMake)
    - Set solution configuration to “Release”
    - Build solution
- Install lcm
    - Open Visual Studio 2017 as administrator
    - Open …\lcm-1.4.0\build\lcm.sln
    - Build the “INSTALL” project
        - Right click the INSTALL project → Project Only → Build Only INSTALL
        - This will install the built binaries into C:\Program Files\lcm
    - Add C:\Program Files\lcm\bin to the system PATH environment variable

Building LCM from Source (Linux)

(Tested on clean Ubuntu Mate 18.04.1 LTS 64-bit virtual machine)

Install prerequisites

- Java OpenJDK and JavaFX

```
sudo apt install openjdk-8-jdk
sudo apt install openjfx
```

- Set JAVA_HOME environment variable
    - Open /etc/profile with sudo priviledges and add the following line at the bottom

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

    - Log out and log in again
- Install pip

```
sudo apt install python-pip
```

- Cython (C-Extensions for Python)

```
pip install cython
```

- Pyjnius Python package

```
pip install pyjnius
```

- Git

```
sudo apt install git
```

- CMake

```
sudo apt install cmake
```

- GLib2

```
sudo apt install libglib2.0-dev
```

- LCM
    - Download and extract LCM from v1.4.0 (tar.gz) from https://github.com/lcm-proj/lcm/releases
    - Open terminal in extracted lcm-1.4.0 directory and run

```
mkdir build
cd build
```

```
cmake ..
make
sudo make install
```