

ML5 mit Eclipse CDT (JH)

Variante 1 (Shared Library)

Vorbereitungsmaßnahmen

1. Unter *Window - Preferences - C/C++ - File Types* prüfen, ob die Dateierweiterung (Pattern) `*.inl` dem Dateityp "C++ Header File" zugeordnet ist.
2. Die Umgebungsvariable `MINILIB_HOME` auf `~minlib/ml5/product` setzen.
3. Umgebungsvariable `PATH` um `%MINILIB_HOME%\build\Eclipse\minilib\Debug` erweitern.
4. Die Umgebungsvariable `MINGW_HOME` muss entsprechend gesetzt sein.

Wichtig: Umgebungsvariablen setzen *bevor* Eclipse gestartet wird.

`libminilib.a` erzeugen

1. Ein neues C++ Projekt `minilib` im Verzeichnis `%MINILIB_HOME%\build\eclipse/minilib` erzeugen (es reicht die Debugversion).
2. Pfad-Variable in Eclipse definieren: *Properties -> Resource -> Linked Resources -> New:*

- *Name:* `MINILIB_HOME`
- *Location:* `${PROJECT_LOC}/../../../../`

3. *Properties - C/C++ Build - Settings*

1. *Tool Settings:* die Compileroptionen

- `-std=c++17,`
- `-pedantic,`
- `-Wall,`
- `-Wextra` und
- `-Wconversion`

definieren.

2. *Build Artifact: Shared Library* auswählen

4. *Properties - C/C++ General - Paths and Symbols*

1. *Includes:* Für *GNU C++* die Pfade

- `/minilib/inc` (*Is a workspace path*)
- `${MINGW_HOME}/include/wx-3.0,`
- `${MINGW_HOME}/lib/wx/include/msw-unicode-3.0`

definieren und *exportieren* (Durch das Exportieren werden sie in Projekten, welche die Minilib referenzieren, automatisch hinzugefügt).

2. *Symbols*: die Makros

- `__WXMSW__`
- `GLFW_DLL`
- `WXUSINGDLL`
- `_FILE_OFFSET_BITS=64`

definieren und *exportieren*.

1. alle Symbole *exportieren*

3. *Libraries*: die Bibliotheken

- `glu32,`
- `opengl32,`
- `wx_baseu-3.0,`
- `wx_mswu_core-3.0`
- `wx_mswu_gl-3.0`

eintragen und *exportieren*.

4. *Source Location*:

- *Link Folder*:
 - `MINILIB_HOME/src`
 - `MINILIB_HOME/inc`
- *Vorgehensweise*:
 - *Link to folder in the file system* selektieren
 - *Variables* -> Pfad-Variable `MINILIB_HOME` auswählen
 - *Extend* -> `{src, inc}`
- Bestehenden *Source Folder* `/minlib` löschen

5. Die DLL `libminilib.dll` wird im Verzeichnis `%MINILIB_HOME%\build\Eclipse\minilib\Debug` erzeugt, das sich im Ausführungspfad befinden muss.

ML5-Applikation `draw` erzeugen

1. Ein neues C++ Projekt `draw` im Verzeichnis `%MINILIB_HOME%\build\eclipse/samples/draw` erzeugen (es reicht die Debugversion).

2. Pfad-Variable in Eclipse definieren: *Properties* -> *Resource* -> *Linked Resources* -> *New*:

- *Name*: `MINILIB_HOME`
- *Location*: `${PROJECT_LOC}/../..../..`

3. *Properties* - C/C++ *Build* - *Settings*

1. *Tool Settings*: die Compileroptionen

- `-std=c++17,`
- `-pedantic,`

- `-Wall`,
- `-Wextra` und
- `-Wconversion`

definieren.

4. *Properties - C/C++ General - Paths and Symbols*

1. *References*: das Projekt `minilib` selektieren -> *Apply*. Die im Library-Projekt exportierten *Symbole* und *Include-Pfade* werden automatisch übernommen. Den Include-Pfad `/minilib` kann man löschen (`/minilib/inc` reicht aus).
2. *Symbols/Includes/Libraries*: Von `minilib` exportierte Symbole, Include-Pfade und Bibliotheken werden automatisch übernommen.

- Überflüssiger Include-Pfad `/minilib` kann gelöscht werden.

3. *Source Location*:

- *Link Folder*: `MINILIB_HOME/samples/draw/src`
- Bestehenden *Source Folder* `/draw` löschen

5. Die Applikation `draw.exe` wird im Verzeichnis

`%MINILIB_HOME%\samples\draw\build\Eclipse\minilib\Debug` erzeugt.

Variante 2 (Static Library)

Vorbereitungsmaßnahmen

1. Unter *Window - Preferences - C/C++ - File Types* prüfen, ob die Dateierweiterung (Pattern) `*.inl` dem Dateityp "C++ Header File" zugeordnet ist.
2. Die Umgebungsvariable `${MINGW_HOME}` muss entsprechend gesetzt sein.

`libminilib.a` erzeugen

1. Ein neues C++ Projekt `minilib` im Verzeichnis `~minilib/ml5/product/build/eclipse/minilib` erzeugen (es reicht die Debugversion).
2. Pfad-Variable in Eclipse definieren: *Properties -> Resource -> Linked Resources -> New*:

- *Name*: `MINILIB_HOME`
- *Location*: `${PROJECT_LOC}/../..`

3. *Properties - C/C++ Build - Settings*

1. *Tool Settings*: die Compileroptionen `-std=c++17`, `-pedantic`, `-Wall`, `-Wextra` und `-Wconversion` angeben
2. *Build Artifact*: "Static Library" auswählen

4. *Properties - C/C++ General - Paths and Symbols*

1. Includes:

1. Für GNU C++ die Pfade `/minlib/inc` (Workspace), `${MINGW_HOME}/include/wx-3.0` und `${MINGW_HOME}/lib/wx/include/msw-unicode-3.0` definieren
2. Alle Include-Pfade *exportieren*. Damit werden sie in Projekten, welche die Minilib referenzieren, automatisch hinzugefügt.

2. Symbols:

1. die Makros `__WXMSW__`, `GLFW_DLL`, `WXUSINGDLL`, `_FILE_OFFSET_BITS=64` definieren
2. alle Symbole *exportieren*

3. Source Location:

- *Link Folder:*
 - `MINILIB_HOME/src`
 - `MINILIB_HOME/inc`
- *Vorgehensweise:*
 - *Link to folder in the file system* selektieren
 - *Variables* -> Pfad-Variable `MINILIB_HOME` auswählen
 - *Extend* -> `{src, inc}`
- Bestehenden *Source Folder* `/minlib` löschen

5. Die statische Bibliothek `libminilib.a` wird im Verzeichnis `~minilib/ml5/product/debug` erzeugt.

ML5-Applikation **draw** erzeugen

1. Ein neues C++ Projekt **draw** im Verzeichnis `${MINILIB_HOME}/ml5/product/build/eclipse/samples/draw` erzeugen (es reicht die Debugversion).
2. Pfad-Variable in Eclipse definieren: *Properties* -> *Resource* -> *Linked Resources* -> *New*:
 - *Name*: `MINILIB_HOME`
 - *Location*: `${PROJECT_LOC}/../..`
3. *Properties* - C/C++ *Build* - *Settings*
 1. *Tool Settings*: die Compileroptionen `-std=c++17`, `-pedantic`, `-Wall`, `-Wextra` und `-Wconversion` angeben
4. *Properties* - C/C++ *General* - *Paths and Symbols*
 1. *References*: das Projekt `minilib` selektieren -> *Apply*. Die im Library-Projekt exportierten *Symbole* und *Include-Pfade* werden automatisch übernommen. Den Include-Pfad `/minilib` kann man löschen (`/minilib/inc` reicht aus).
 2. *Libraries*: die Bibliotheken `glu32`, `opengl32`, `wx_baseu-3.0`, `wx_mswu_core-3.0` und `wx_mswu_gl-3.0` eintragen (`minilib` wird automatisch übernommen)

3. Source Location:

- Link Folder: `MINILIB_HOME/samples/draw/src`
- Bestehenden Source Folder `/draw` löschen

5. Die Applikation `draw.exe` wird im Verzeichnis `~minlib/ml5/product/samples/draw/debug` erzeugt.