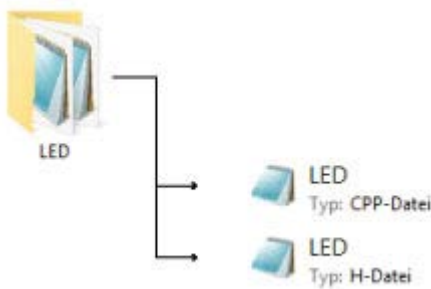


## Was sind Bibliotheken?

Bei der Installation der Arduino-Software wird Windows-Rechnern unter *C:/Programme(x86)* ein Ordner mit Namen „Arduino“ erstellt. Darin befinden sich neben den verschiedenen Programmen und Konfigurationsdateien weitere Unterordner. Einer davon heißt „libraries“. In ihm werden die Bibliotheksordner abgelegt, die mit der Software ausgeliefert werden. Welche Bewandtnis das hat, soll im Folgenden erklärt werden. Dazu wird auf eine Bibliothek mit Namen „LED“ genutzt, die eigens für diese Erklärung konzipiert worden ist. Den Hintergrund bildet das Programm „Blink“, das allen Arduino-Interessierten bekannt sein dürfte.

Der Bibliotheksordner „LED“:



Soll ein Ordner von der Arduino-Software als Bibliothek erkannt werden, müssen sich in ihm zwei Dateien des Typs .h und .cpp befinden und ihre Namen müssen mit dem des Ordners übereinstimmen (siehe Abbildung links).

Die Datei „LED.h“:

```
LED - Editor
Datei Bearbeiten Format Ansicht ?
#ifndef LED_H
#define LED_H
#include "Arduino.h"
class LED
{
    public:
        LED();
    private:
};
#endif
```

Im Textfenster eines Editors zeigt sich die Datei wie links abgebildet

Die if-Bedingung zu Beginn des Codes soll einen Abbruch bewirken, wenn keine Bibliothek mit der Bezeichnung „LED“ gefunden wird.

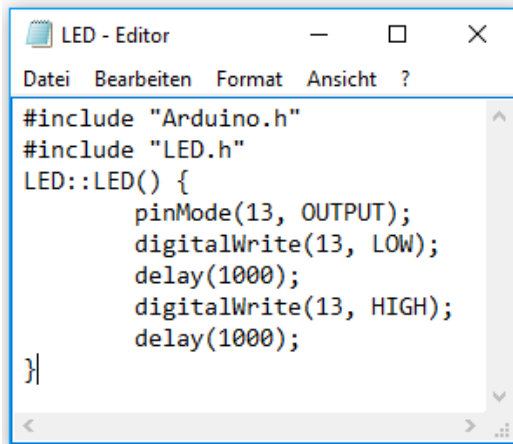
Der #define-Befehl stellt sicher, dass nur diese Bibliothek genutzt wird.

Über den #include-Befehl zusammen mit der Datei „Arduino.h“ gewährleistet die Passung zur Arduino-Software.

Mit „class LED“ wird ein Kapitel („class“) mit der Bezeichnung „LED“ geöffnet, in das unter „public“

Methoden und unter „private“ Variablen eingebunden werden können. In diesem speziellen Fall enthält das Kapitel nur eine Methode mit der Bezeichnung „LED();“.

Die Datei „LED.cpp“:

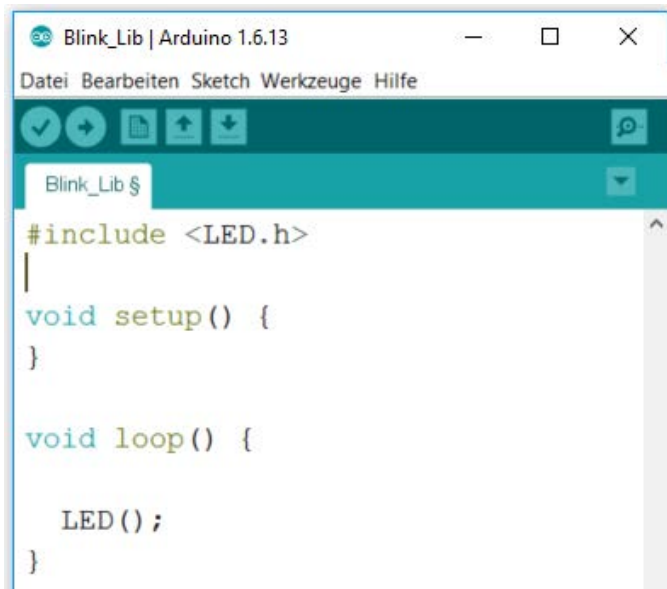


```
#include "Arduino.h"
#include "LED.h"
LED::LED() {
    pinMode(13, OUTPUT);
    digitalWrite(13, LOW);
    delay(1000);
    digitalWrite(13, HIGH);
    delay(1000);
}
```

In der CPP-Datei wird die im LED-Kapitel festgelegte Methode „LED();“ mit Programmcode gefüllt, hier dem Teil, der nötig ist, um eine LED im Sekundentakt ein- und auszuschalten. Die beiden Befehle am Anfang stellen sicher, dass die Arduino-Vorgaben (#include „Arduino.h“) ebenso eingehalten werden wie die aus der H-Datei (#include „LED.h“).

Soll nun mit Hilfe dieser Bibliothek eine LED an Pin 13 zum Blinken gebracht werden, genügt es in der Arduino-IDE mit der Tastenkombination *Strg+N* eine neue Datei anzulegen, über *Sketch > Bibliothek*

*einbinden > Contributed Bibliotheken* LED anzuwählen und unter „void loop“ die Methode „LED();“ einzufügen.



```
#include <LED.h>
|
void setup() {
}

void loop() {

    LED();
}
```

Wird dann der Code auf den Arduino übertragen, fängt die eingebaute LED im Sekundentakt an zu blinken.

Wer das Beispiel erproben möchte, kann die „LED library“ unter [www.mint-unt.de/downloads.html](http://www.mint-unt.de/downloads.html) als ZIP- Datei herunterladen und auf dem üblichen Weg einbinden: *Sketch > Bibliothek einbinden > .ZIP-Bibliothek hinzufügen*. Auch kann die ZIP-Datei konventionell entpackt und der Bibliotheksordner unter *C:/Programme(x86)/Arduino/libraries* gespeichert werden.