
Darstellung der Projektergebnisse

Projekt: *StreetLight*
Dokument: *Dokumentation StreetLight*
Version: *1.0*
Datum: *19.01.2020*
Status: *Dokumentenstatus: verabschiedet*

Zusammenfassung

Beim StreetLight Projekt handelt es sich um eine Arduino-bibliothek, die die Kontrolle von Ampeln und ähnlichem erlaubt über einen Arduino, entweder mit Led-Lichtern oder nach geringen Modifikationen anderen Leuchtmitteln. Die Bibliothek ist anwenderfreundlich und enthält Beispiele für verschiedene Anwendungsarten, sowie ein fertiges Programm für das Steuern der Ampel.

1 Benutzerdokumentation

Wie wird das Projekt in Betrieb genommen?

Man benötigt einen Arduino mit entsprechend des Schaltplans verbundenen Schalter und Lichter zur initialen Inbetriebnahme, die genauen Pins sind nicht wichtig, solange man diese im Programm dann auch ändert, später ist es nur wichtig irgendeine Art von out und Input zu haben, die die Light.h und Button.h Headerdateien erfüllen können mit den entsprechend umgeschriebenen .cpp Dateien.

Was ist zu tun, um das Projekt zu verwenden?

Man muss einen Arduino an einen entsprechenden Schaltkreis anschließen und am besten das setup_example durchlesen um die meiste Funktionalität der Bibliothek für den leihen zu verstehen, in den verschiedenen Headerdateien stehen zusätzliche Funktionalitäten und Erweiterungen. Zum generellen nutzen ist nur ein Setup und der Aufruf des Programmes wie in dem angesprochenen Beispiel von Nöten. Für fortgeschritten Nutzen mit anderen Lichtsystemen als Leds oder anderen Inputs als Taster ist ein Umschreiben der Light.cpp und Button.cpp von Nöten.

Wie mache ich einen Eingangstest?

Für einen Eingangstest steht das hardware_test_sketch unter den Beispielen im example Ordner zur Verfügung.

Welche Funktionalität steht durch welche Bedienung zu Verfügung?

Beim Hauptprogramm der Library handelt es sich um eine Ampelschaltung, die beim einschalten sich in einem Standby-modus befindet, in welchem ein Licht blinkt, naturgetreu wäre das das gelbe, und auf tasten druck mit dem Ampelzyklus beginnt. Dieser wird durch ein erneutes kurzes drücken entweder gestoppt oder wieder fortgesetzt oder durch ein Langes drücken des Knopfes beendet um wieder beim Blinken des Standby-modus zu landen. Dieser wird wiederum nach einer bestimmten Zeit ohne weitem Tasteninput beendet und die Ampel schaltet sich aus. Sie kann durch erneuten Tastendruck wieder angeschaltet werden und der

Standby-modus läuft wieder. Die Bibliothek bietet darüber hinaus bis zu fünf Ampelphasen, die in einer schleife ausgeführt werden, zu erstellen und diese mit bis zu drei Lampen und eigener Dauer zu deklarieren. Darüber hinaus können insgesamt bis zu fünf Lampen verwendet werden, welche entweder ein oder zwei Leds steuern um eine Ampel an zwei Seiten einer Kreuzung zu spiegeln. Andere Parameter können ebenfalls eingestellt werden, wie die Blinkgeschwindigkeit. Letztlich sind die Light.cpp und Button.cpp Dateien einfach für andere arten Hardware modifizierbar.

Bei der Benutzung der Bibliotheken ist darauf zu achten, wenn man die Phasen deklarier, man keine auslässt und die Höchstzahl auch wirklich auf die Letzte Setzt, da es sonst zu undefiniertem Verhalten kommen kann.

Am einfachsten ist es das Programm einfach laufen zulassen wie in Setup_example , man kann das Programm aber auch manuell steuern wie im support_example wo die Bibliothek nur als support dient und der Sketch die Hauptarbeit des Programmes macht.

Wie setze ich das Projekt außer Betrieb?

Einfach Stecker, nichts besonderes zu beachten.

2 Entwicklerdokumentation

Hier beschrieben sind die Commits bis zu Commit „““““ im Github-Repository „“““““““

Alle unterlagen wie Schaltpläne befinden sich dort.

Da es sich um ein Arduino-Projekt handelt Läuft die Code Erstellung für die Sketches und die Übertragung auf Hardware über die Arduino IDE, die Bibliothek wurde über Notepad++ erstellt.

Zum Überprüfen ob die Programme noch funktionieren sollte nachdem die Hardware getestet wurde über das hardware_test_sketch, die anderen Beispiele getestet werden. Alle drei sollten ungefähr die gleiche Funktionalität haben, wie in der Benutzerdokumentation beschrieben, wobei der functional_example_without_library Sketch sich auch im pause Modus abschaltet nach einer gewissen zeit. Sollte ein Fehler Vorliegen ist es vielleicht hilfreich im setup_example die runLight() Methode durch den auskommentierten Code oder runLightIntern() zu ersetzen.

3 Bestehende Probleme

Es muss darauf geachtet werden, das keine Pointer auf nicht existierende Objekte entstehen, man darf beim deklarieren der Phasen keine Überspringen und man darf die MaxPhase nicht zu hoch setzten. Ebenfalls darf bei setFlashlight() kein Index einer nicht deklarierten Lampe benutzt werden.

4 Weiterentwicklung

Weitere Absicherungen, wie in den Bestehen Problemen beschrieben, ausführlichere Kommentierung der .cpp Dateien und Vereinfachung des Codes, an manchen Stellen gibt es noch Codeverdoppelungen und weniger als ideale Lösungen, vor allem bei den Funktionen, die im support_example benutzt werden. Weiter mögliche Weiterentwicklungen ist die Erweiterung für

mehr Phasen und Lichter, sowohl insgesamt als auch pro Phase. Ebenfalls möglich ist das hinzufügen von Funktionszeiger zu dem Konstruktor für die Phasen oder der lightOperation() Methode um während bestimmten Phasen noch ein zusätzliches Warnlicht blinken zu lassen oder andere Zusatzfunktionalität zu ermöglichen.

Man kann auch schon weitere Light.cpp und Button.cpp Beispiele für unterschiedliche Hardware erstellen.