**Das GBPP - Geraffel-Banner-Pattern Protokoll**

Damit jeder seiner Kreativität freien Lauf lassen kann, wurde ein einfaches System implementiert, nach dem die Lichtmuster angelegt werden können. Dies geschieht im Moment noch statisch durch Übernahme in das jeweilige Programm-Template.

Auf dem Camp können wir gerne die Möglichkeit eines Uploads über das Web-Interface diskutieren und über Möglichkeiten zu automatisierten (ggf. zufälligen) Lichtmusterauswahl, z.B. bei Inaktivität auf dem Web-Interface, nachdenken.

Das GBPP sieht vor, dass jede Szene eines Lichtmuster in einer separaten Zeile einer herkömmlichen Text-Datei angeordnet wird. In dieser Zeile wird also der Zustand aller acht LED-Buchstaben, also deren Helligkeit, die Dauer (wie lang wird die Szene angezeigt) und die Übergangsvariante zur nächsten Szene abgelegt. Geübte GBPP (hier: Geraffel-Banner Programming-Professionals) haben dann noch die Möglichkeit über das sogenannte „Smooth-Byte“ anzugeben, in welcher Abstufung der Übergang zum nächsten Lichtmuster erfolgen soll.

Die Werte werden durch Kommata getrennt.

Im Gegensatz zum ursprünglichen Geraffelbanner, bei dem die einzelnen Buchstaben nur ein- oder ausgeschaltet werden konnten, ist in der neuen Version eine Helligkeitsreglung über Pulsweiten-Modulation möglich (OK, eigentlich werden sie auch nur ein- und ausgeschaltet, aber viel schneller und mit variabler Pulsbreite), das bietet eine zusätzliche Dimension bei der Gestaltung der Lichtmuster, da jeder Buchstabe zu jeder Phase des Programmablaufs eine individuelle Helligkeit haben kann.

Der Übergang zur Folgeszene, welche in der jeweils nächsten Zeile steht kann auch fließend geschehen, indem man über eine adäquate Wahl des „Smooth-Bytes“ eine sanfte Überblendung ermöglicht. Die Angabe „1“ stellt einen „harten“ Übergang dar (s.u.).

Die Parameter haben wie folgt auszusehen:

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Wert€ | Wertebereich | Kommentar |
| Helligkeitswert | 1 – 8 | 0 – 99 | siehe unten |
| Pausenzeit | 9 | 0 – 10000 | (in Millisekunden = 0 -10sek.) |
| Smooth-Byte | 10 | 1 – 99 | 1=direkter Übergang nach Ablauf der Pausenzeit |
|  |  |  |  |

Die Werte 1 bis 8 entsprechen dabei den Buchstaben auf dem Banner

1=**G**, 2=**E**1, 3=**R**, 4=**A**, 5=**F**1, 6=**F**2, 7=**E**2, 8=**L**

hier ein Beispiel:

1,99, 1,99, 1,99, 1,99, 3000, 20,

99, 1,99, 1,99, 1,99, 1, 3000, 20,

1,99, 1,99, 1,99, 1,99, 3000, 20,

99, 1,99, 1,99, 1,99, 1, 3000, 20,

99, 1,99, 1,99, 1,99, 1, 1000, 20,

50,50,50,50,50,50,50,50, 3000, 20,

50,50,50,50,50,50,50,50, 2000, 1,

1, 1, 0, 0, 0, 0, 1, 1, 3000, 8,

1, 1, 0, 0, 0, 0, 1, 1, 2000, 20,

1, 2, 4, 8,16,32,64,99, 3000, 30,

1, 2, 4, 8,16,32,64,99, 1000, 1,

99,64,32,16, 8, 4, 2, 1, 3000, 30,

99,64,32,16, 8, 4, 2, 1, 1000, 20,

99,99,99,99,99,99,99,99, 3000, 1,

99,99,99,99,99,99,99,99, 2000, 40,

**Tripps & Ticks**

* Das Lichtmuster wird nur einmal „durchgefahren“, sollen sich Passagen oder das ganze Muster wiederholen, so muss der betreffende Teil kopiert und ein/angefügt werden. (Hierzu sollten wir weitere Optionen überlegen, z.B. Auto-Repeat bis neues Pattern aufgerufen wird))
* ist ein weicher Übergang zwischen zwei Lichtmustern gewollt ist das Smooth-Byte relativ hoch einzustellen. Wenn jedoch die Folgewerte des gesamten Musters sich kaum von den aktuellen Werten unterscheiden ist das nicht sinnvoll.
* Faustformel: Smooth-Byte = maximal so groß wie die größte Differenz der 8 Helligkeitswerte (gemeint ist die Differenz der Helligkeit je jeweiligen Buchstaben vom aktuellen Wert auf den angedachten Folgewert)
* Ist das Smooth-Bit „1“ bedeutet dies einen direkten Wechsel auf den Folgewert nach Anlauf der Pausenzeit
* Bei der Dimensionierung der Helligkeitswerte ist die Physiologie des Auges zu berücksichtigen. Dies bedeutet in der Praxis, dass benachbarte Werte für großer Helligkeit der LEDs optisch kaum wahrgenommen werden können. Der Sprung von Helligkeitswert 1 auf 3 ist deutlich zu erkennen, von 90 auf 99 aber kaum.
* Das bedeutet, Programme die ausschließlich Werte > 50 haben werden eher als „Wabern“ wahrgenommen, speziell wenn das Smooth-Bit noch viele Zwischenschritte vorgibt
* Weiterhin sollte das Smooth-Bit bei sehr kurzen Pausenzeiten ebenfalls klein gehalten werden, da sich z.B. ein Überblenden mit Smooth-Bit 30 bei einer Pausenzeit von 100ms optisch überhaupt nicht auflösen und wahrnehmen lässt
* Eine Pausenzeit < 100 ist kaum mehr sinnvoll, da damit das aktuelle Muster beinahe sofort übersprungen würde
* Das letzte Lichtmuster eines Programmes blendet nach der entsprechenden Pausenzeit wieder in das erste Muster über, danach ist der Programmablauf beendet.