

Einsatzanleitung

Das Projekt hat die Zielsetzung auf einem sich drehenden Display alle Art von Information anzeigen zu können. Als Demonstration dafür wurden ein Uhrzeit-, Datums-, Temperatur- und Luftfeuchtigkeitsanzeige implementiert. Diese sind standardmäßig aktiviert und werden während des Betriebs ständig gewechselt, ohne dass eine Eingabe durch den Benutzer erforderlich ist.

Zum Betrieb muss die Schaltung laut der Aufbauanleitung aufgebaut und in Drehung gegen den Uhrzeigersinn versetzt werden. Durch den Hallsensor wird dann automatisch die Geschwindigkeit ermittelt und der jeweilige Displaybuffer ausgegeben.

1.: Inbetriebnahme:

1.1:Erster Start:

Zur erfolgreichen Inbetriebnahme müssen zunächst laut Aufbauanleitung alle Komponenten verbunden werden.

Anschließend muss der richtige Anpressdruck und Kontakt der Kabelbürsten auf der Kupferlauffläche des Motors eingestellt werden. Dazu muss die Haltescheibe gelockert und die Kabel an eine passende Position geschoben werden. Je nach verwendetem Kondensator genügt bereits ein leichter Druck auf die Laufflächen.

Dann muss die Fernsteuerung eingeschaltet und der geladene Akku an den Regler angeschlossen werden. Der Motor bestätigt mithilfe eines Piepcodes seine Funktionsbereitschaft und kann über den Gashebel gesteuert werden (Regler an Kanal 3 des Empfängers). Zunächst sollte nur mit niedriger Drehzahl gefahren werden. Nun können eventuell auftretende Vibrationen durch Unwucht ausgeglichen werden. Dazu wird frei nach dem Prinzip „trial and error“ zunächst auf einer Seite ein Gewicht bestehend aus Klebeband aufgebracht und überprüft, ob die Vibrationen zu- oder abnehmen. Dann wird entsprechend mehr Klebeband hinzugefügt oder die Seite gewechselt.

Anschließend muss der Aufbau bis die Lampe am GPS-Modul nicht mehr blinkt stehen, (ca. 2 - 3 Minuten) da die Zeit der dort eingebauten Echtzeituhr mit einem GPS-Satelliten synchronisiert werden muss.

Nun ist das System bereit für den normalen Betrieb

1.1:Normaler Betrieb:

Zum normalen Betrieb nach dem ersten Start wird wie in 1.1 beschrieben zunächst der Sender und dann der Aufbau mit Spannung versorgt. Dann kann langsam beschleunigt werden, bis ein stabiles Bild angezeigt wird. Dies ist normalerweise bei etwa 20% des Gasausschlags der Fall.

2.: Informationen zur Erstellung neuer Funktionalität mithilfe der eingebauten Funktionen

2.1.: Aufbau des Displaybuffers:

Im Folgenden soll der Aufbau die nutzbaren Funktionen zur Veränderung des Displaybuffers erläutert werden, mit dem das Hinzufügen eigener Funktionen zu der Gegebenen Grundfunktionalität sehr einfach zu realisieren ist.

Der Displaybuffer besteht aus einem Globalen, 3 dimensionalen Array mit einstellbarer Breite. Höhe und Tiefe sind durch die Anzahl der LEDs (8) und die Farben Rot, Grün und Blau (3) vorgegeben. Die Spalten werden nacheinander ausgegeben und sind wie folgt formatiert:

{G,R,B}	{G,R,B}	{G,R,B}	{G,R,B}	{G,R,B}	{G,R,B}	{G,R,B}	{G,R,B}
Untere LED				Obere LED			

2.2.: Eingebaute Funktionen zur Manipulation des Buffers:

- **void byteToRGB(byte val, int start, int r, int g, int b)**
Mit dieser Funktion kann ein ganzes Byte in der Farbkombination r,g,b an Position start des Displaybuffers gelegt werden. Damit können z.B. eigene Zeichen sehr leicht realisiert werden.
Verwendungsbeispiel: byteToRGB(0B11110000,0,255,255,255);
Färbt in der ersten angezeigten Spalte die Untere Hälfte weiß
- **void monoToRGB(int letter, int start, int r, int g, int b)**
Diese Funktion fügt einen 5 spalten breiten, einfarbigen Buchstaben aus der vorhandenen Ascii.h nach der Startposition in der Farbkombination r,g,b in den Buffer ein.
Verwendungsbeispiel: monoToRGB('A',0,255,255,255);
Zeigt in den ersten 5 Spalten ein weißes A an.
- **void strToAry(char *s, int start, int r, int g, int b)**
Die Funktion verwendet monoToRGB und fügt einen ganzen String in den Displaybuffer ein.
Verwendungsbeispiel: strToAry("Hallo Welt", 0,255,255,255);
Zeigt den Text „Hallo Welt“ beginnend an der Stelle 0 an