VOR-Projekt

Team Nordsignal

Anforderungen:

Im Rahmen des VOR Projektes sollte ein Programm entwickelt werden, welches in regelmäßigen zeitlichen Abständen ein Nord Impuls sendet.
Zu Beginn des Projektes wurde die Frequenz auf 1Hz festgelegt, jedoch später zu 60Hz zugunsten anderer Teams verändert.
Das Senden des Signals sollte über einen normalen, noname, 315/433 MHz Sender geschehen.

Problematik:

Dies führte zu massiven Problemen in folgenden Bereichen: Die Sendeleistung der Sender war nicht ausreichend um eine einfache Bitfolge zu senden, da in den Zeiträumen in welchen das Signal NULL ist, viele Störsignale empfangen werden.
Dies fiel aber erst auf, als versucht wurde den Nord Impuls mit einem Arduino, welcher mit einem entsprechenden Empfänger ausgestattet wurde, auszuwerten.
Alle Messungen am Oszilloskop wiesen auf eine Problemfreie funktionsweise hin.
Dies liegt an der, im Oszilloskop integrierten, Filterung.

Lösungsansatz #1:

Eine, vermeintliche Lösung wäre das Arduino Wireless SD Shield, welches über den Serial Port des Arduinos verbunden werden kann.
Das Wireless SD Shield bietet die Möglichkeit zur kabellosen Übertragung ohne zusätzliche Hardware (außer das Wireless SD Shield) und zusätzlichen Code für kabellose Übertragung.
Das Wireless SD Shield kann genau wie jede andere USB Schnittstelle angesprochen werden.
Die Idee des Wireless SD Shield wurde verworfen, da dieses vergleichsweise kostenintensiv ist (ca 20€).

Lösungsansatz #2:

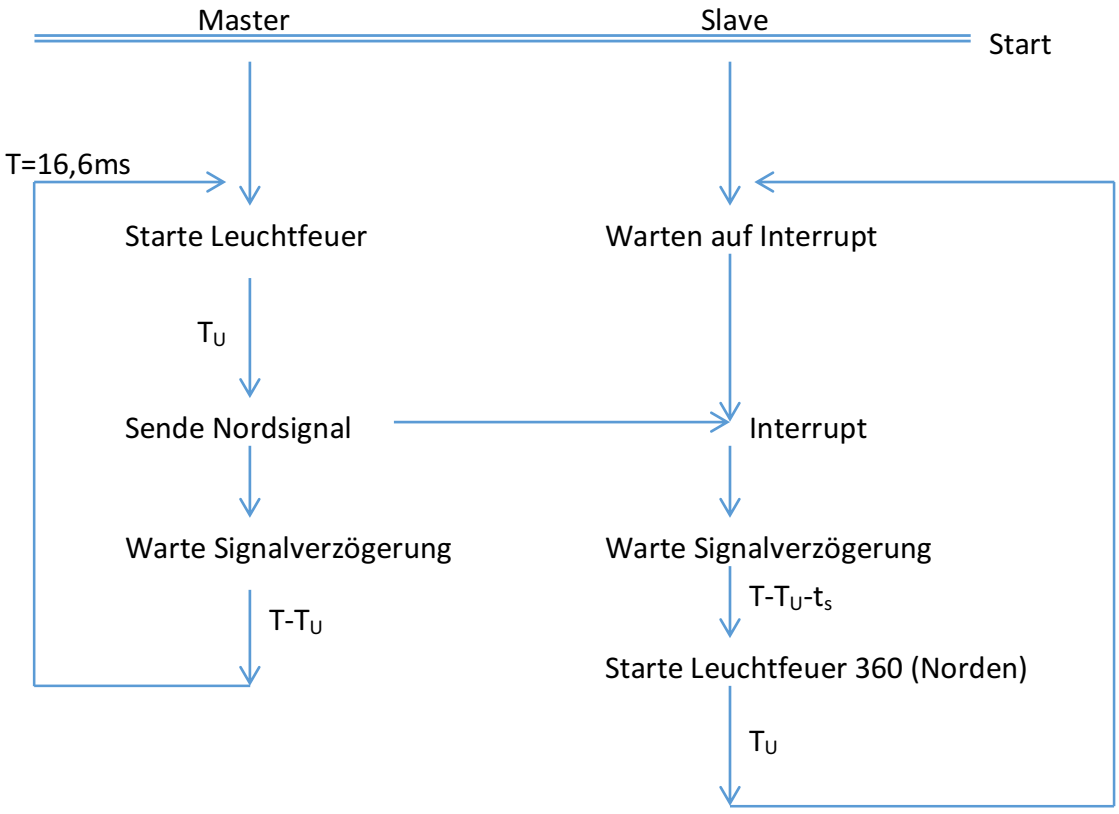
Mit Hilfe einer öffentlich zugänglichen Library, welche verwendet werden kann um RC Verbindungen zu simulieren, kann ein weniger empfindliches Signal gesendet werden.
Wenig problematisch ist das Modulieren und Senden des Signals. Empfangen ist Auswerten ist mit mehr Schwierigkeiten verbunden.

Die dafür verwendete Bibliothek kann auf https://github.com/sui77/rc-switch gefunden werden.

Synchronisation:

- Serielle Kommunikation bei ARDUINO funktioniert sehr einfach und problemfrei
- Synchronisation nach Master-Slace-Prinzip bei jeder Umdrehung
- Benutzung von Interrupts am Slave

- Messung der Laufzeit vom Senden des Nordsignals und bearbeiten des Interrupts bis davon ausgelöster Befehl startet
- Nach Umdrehung muss eine etwas größere Zeit gewartet werden um konstante Periodendauer einzuhalten
- $f_{CPU} = 16MHz$



Senden:

Es existieren 3 sinnvolle Methoden zum Übertragen von Strings.

Methode #1

Die einfachste Methode übermittelt einen einfachen Binärstring.

```
#include <RCSwitch.h>

RCSwitch mySwitch = RCSwitch();

void setup() {
  mySwitch.enableTransmit(10);  // Using Pin #10
}

void loop() {
  mySwitch.send("000000000001010100010001");
  delay(1000);
```

```
}
```

Methode #2

Übertragen von Dezimal Werten.

```
#include <RCSwitch.h>

RCSwitch mySwitch = RCSwitch();

void setup() {
  mySwitch.enableTransmit(10);  // Using Pin #10
}

void loop() {
  mySwitch.send(5393, 24);
  delay(1000);
}
```

Methode #3

Übertragunsmethode zum Senden von Tristates.

```
#include <RCSwitch.h>

RCSwitch mySwitch = RCSwitch();

void setup() {
  mySwitch.enableTransmit(10);  // Using Pin #10
}

void loop() {
  mySwitch.sendTriState("00000FFF0F0F");
  delay(1000);
}
```

Fazit:

In einer kontrollierten Umgebung funktionieren alle drei Methoden gleichermaßen gut.
Es ist lediglich darauf zu achten, dass der Empfänger auch auf das richtige Signal horcht, ansonsten ist eine Übertragung nicht möglich.
Außerhalb der kontrollierten Umgebung wurde nicht mehr jeder Nord Impuls erkannt.
Als Reaktion darauf sollten die Slave Türme weiterdrehen und sich zurücksetzen sobald sie einen erneuten Impuls empfangen. Für die Fahrzeuge sollte der Nordimpuls Timer verändert werden, damit dieser nicht mit 60 Hz zurückgesetzt werden muss.

Eine gut ausgelegte Filterung der Signale von Seitens der Empfänger wäre die logische Konsequenz.