

# GPS-Software Decoder

---

Von: Laborgruppe 24, bestehend aus

Nicolas Lauinger 44343, Manuel Härtle, 46866

## Aufgabenstellung:

Aus einem Summensignal soll herausgearbeitet werden welcher der 24 verfügbaren GPS Satelliten mit welcher Verschiebung der Bitsequenzen für die Generierung des Signals verantwortlich ist.

## Lösungsstrategie

1. Einlesen des Summensignals
2. Generieren der Chipsequenzen für alle Satelliten
3. Kreuzkorrelation des Summensignals mit allen Chipsequenzen in allen möglichen Permutationen
4. Erkennen von Peakwerten der Kreuzkorrelationen

## Umsetzung

Im ersten Schritt wird ein Dateiname einer einzulesenden Datei per Konsoleneingabe übergeben. Dieser Dateiname wird im Anschluss an die Decoder-Klasse übergeben, die den Parameter wiederum an SumSignal übergibt.

In SumSignal wird nun eine Referenz auf eine Datei erzeugt, sodass diese im Anschluss ausgelesen werden kann. Die Datei ist leerzeichensepariert aufgebaut. Daher kann die Filtereigenschaft von `fscanf` ausgenutzt werden um die einzelnen Summenbits des Summenstreams zu splitten und diese anschließend in ein Array zu schreiben. Die eingelesenen Daten entsprechen dem später benötigten Summensignal der 4 GPS Satelliten.

Die `decode`-Methode der Decoder-Klasse stößt außerdem die Generierung der Chipsequenzen an. Dies geschieht in der Satellitenklasse bei Aufruf des Konstruktors. Mit der Methode `getChipSequence` können die individuellen Chipsequenzen eingebunden werden.

Die Chipsequenzen sind für jeden Satelliten individuell und eindeutig zuordenbar. Die Berechnung der einzelnen Signale basiert auf zwei Mutterfolgen eines Goldcode Generators.

Demnach müssen zwei Mutterfolgengeneratoren erstellt werden. Beide Mutterfolgen entstehen durch die XOR-Verknüpfung spezieller Bits eines 10 Bit Schieberegisters. Mutterfolge 1 entsteht indem man das 10 Bit Schieberegister am niederwertigsten Bit und an Bit 8 (Anfang der Zählung 1) XOR verknüpft und dieses Ergebnis wiederum als höchstwertiges Bit in das Schieberegister einfügt.

Analog zu dieser Arbeitsweise wird die zweite Mutterfolge generiert. Hierbei werden jedoch statt dem ersten und achten Bit die Bits 2, 3, 5, 8 und 9 XOR verknüpft.

Konkret bedeutet dies für die erste Mutterfolge: man liest das niederwertigste Bit des aktuellen Registerstatus aus und macht ein XOR auf das achte Bit des aktuellen Registerstatus.

Logisch betrachtet bedeutet dies, man wendet Bitmasken auf die aktuelle Sequenz an. Im ersten Fall wird eine Und Verknüpfung der aktuellen Sequenz mit der Bitmaske  $(1 \ll 0)$  – Binär: 00 0000 0001 – angewendet um das kleinste Bit zu erhalten. Das achte Bit erhält man durch eine Und-Verknüpfung der aktuellen Registersequenz mit der Bitmaske  $(1 \ll 7)$  – Binär: 1000 0000 - . Beide werden nun mit XOR verknüpft, sodass sich der logische Ausdruck  $(sequenz \& (1 \ll 0) \wedge sequenz \& (1 \ll 7) \gg 7)$  ergibt.

Um nun die Sequenz zu shiften und den neuen Wert einzufügen wird ein Rechtsshift angewendet, sodass beispielsweise aus 1000 0000 – 0100 0000 wird und setzt mit Hilfe einer Oder-Verknüpfung das höchstwertigste Bit. Logisch bedeutet dies:  $sequenz = sequenz \mid (neuesBit \ll 9)$ .

Die Chipsequenzen der jeweiligen Satelliten werden erzeugt indem man nun das niederwertigste Bit des aktuellen Schieberegisterzustands der ersten Mutterfolge mit einer Satellitenspezifischen Kombination von Verknüpfungen der zweiten Mutterfolge XOR verknüpft. Für Satellit 1 wird die XOR Verknüpfung von Bit 5 und 9 mit dem niederwertigsten Bit der ersten Mutterfolge XOR verknüpft.

Um die Aufgabe lösen zu können muss nun eine dreifach geschachtelte Schleife aufgebaut und durchlaufen werden. Die höchste Instanz iteriert über die eindeutige Nummer (id) aller Satelliten - also von 0 bis 23 -, die nächste Schleife entspricht der Verschiebung der Chipsequenzen und die innere Schleife ist für die effektive Kreuzkorrelation verantwortlich.

Die Kreuzkorrelation wird durch die Bildung des Skalarprodukts zwischen dem Summensignal und der permutierten Chipsequenz ausgedrückt. Das bedeutet also, dass Komponente 1 des Summensignals mit Komponente 1 der permutierten Chipsequenz multipliziert wird und dieses Ergebnis mit der Verrechnung beider 2. Komponenten addiert wird.

Als Ergebnis erhält man Werte im Intervall -1024 bis 1024. Da in der Regel mit normierten Werten gerechnet wird, werden die Ergebnisse der Kreuzkorrelation durch die Anzahl der Chips (1024) geteilt, sodass sich das Intervall nun von -1 bis 1 erstreckt.

An den Stellen an denen die normierte Kreuzkorrelation den Wert -1 oder 1 hat beginnt jeweils eine Chipsequenz eines Satelliten. Ist der Wert der Kreuzkorrelation -1, so wurde die Chipsequenz eines Satelliten invertiert gesendet, was bedeutet, dass dieser ein 0-Bit gesendet hat. Ist der Wert der Kreuzkorrelation 1, so wurde die Chipsequenz in der originalen Form gesendet, was bedeutet, dass der entsprechende Satellit ein 1 Bit gesendet hat.

Prüft man ausschließlich auf die Peaks -1 und 1, so erhält man nicht alle beteiligten Satelliten. Um dies zu beheben muss ein-Faktor berücksichtigt werden. Dieser resultiert aus der von Robert Gold gefundenen Formel. Die resultierenden Werte sind bei einem 10 Bit Register -65 oder -64. Diese werden mit der Anzahl der beteiligten Satelliten abzüglich dem sendenden (hier 3) multipliziert, sodass sich für diese Implementierung ein Korrekturfaktor von  $3 * 65 = 195$  ergibt.

## Konstanten

### BIT\_1 (1 << 0)

Niederwertigstes Bit des Schieberegisters (der Mutterfolge).

### BIT\_2 (1 << 1)

Zweites Bit des Schieberegisters (der Mutterfolge).

### BIT\_3 (1 << 2)

Drittes Bit des Schieberegisters (der Mutterfolge).

### BIT\_5 (1 << 4)

Fünftes Bit des Schieberegisters (der Mutterfolge).

### BIT\_8 (1 << 7)

Achtes Bit des Schieberegisters (der Mutterfolge).

### BIT\_9 (1 << 8)

Neuntes Bit des Schieberegisters (der Mutterfolge).

### HIGH\_BIT 9

Höchstwertiges Bit des Schieberegisters der Mutterfolge.

### INITIAL\_VALUE 1023

Initialwert zur Initialisierung der Schieberegister zur Generierung der Mutterfolgen. Festgesetzter Wert in diesem Verfahren (11 1111 1111).

### REGISTER\_WIDTH 10

Anzahl der D-Flip-Flops des Schieberegisters zur Generierung der Mutterfolgen. Festgesetzter Wert in diesem Verfahren.

### DELAY

Delay-Faktor, resultierend aus der Sendetoleranz. Wird benötigt um die Toleranz der Autokorrelation zu bestimmen.