

Software-Schnittstelle der Funktionsgeneratoren DDS 10/DDS 110

Mit der Veröffentlichung und Beschreibung der Übertragungsprotokolle der beliebten DDS-Funktionsgeneratoren DDS 10/DDS 110 wird es dem ambitionierten Hobbyprogrammierer ermöglicht, ein eigenes Software-Interface zu programmieren. Dadurch ist es möglich, die Signalgeneratoren über ein simples Konsolenfenster oder über eine an die jeweiligen Bedürfnisse angepasste grafische Programmoberfläche zu bedienen.

Vielseitige DDS-Generatoren

Die beiden programmierbaren DDS-Funktionsgeneratoren DDS 10/DDS 110 sind sehr universell im Frequenzbereich von 0,1 Hz bis 10 MHz einsetzbar. Sie verfügen nicht nur über feine Einstellschritte von 0,1 Hz, eine Wobbel- und Modulations-Funktion und geben Sinus-, Rechteck- und Dreiecksignale ab, sie sind auch vielseitig für die Frequenzaufbereitung, z. B. von PLL-Synthesizern oder KW-Empfängern, einsetzbar.

Beide laufen zur Höchstform auf, wenn die zahlreichen vorhandenen Möglichkeiten der Programmierung und Steuerung durch einen PC genutzt werden. Ein kostenloses Programm hierzu ist im ELV-Internet-Angebot bereits verfügbar. Natürlich liegt es, z. B. im Amateurfunk- und Messtechnik-Bereich, nahe, dass der ambitionierte Nutzer seine eigenen Programm-Applikationen an diese interessanten und vielseitigen Funktionsgeneratoren "anbinden" möchte.

Genau dies soll durch die Veröffentlichung der Software-Schnittstelle möglich gemacht werden. Der folgende Artikel gibt dem Programmierer alle Werkzeuge für die Steuerung der beliebten Funktionsgeneratoren über eine Software-Schnittstelle in die Hand.

Verbindung mit dem Computer

Um eine Datenverbindung zu dem PC herzustellen, muss eine Hardware-Schnittstelle vorhanden sein, worüber die Kommunikation stattfinden kann. Hier gehen die beiden DDS-Boards unterschiedliche Wege.

Das PC-Interface (Schnittstelle)

Das DDS-10-Board ist als Stand-alone-Gerät mit eigenem Display und vollständigen Bedienelementen konzipiert und kann so auch autark arbeiten. Es verfügt über eine serielle UART-Schnittstelle des zentralen Mikrocontrollers. Zur Anbindung an einen PC benötigt dieses Gerät einen zusätzlichen Schnittstellenwandler, um die UART-Schnittstelle des Mikrocontrollers für den Computer nutzbar zu machen. Dabei wird entweder der reelle COM-Port des Computers verwendet oder über USB ein virtueller COM-Port eingesetzt. Welche Schnittstellenwandler genutzt werden können und wie diese anzuschließen sind, ist in der Bedienungsanleitung der DDS-10-PC-Software nachzulesen.

Das ELV-Programm hält dazu zwei geeignete USB-Module sowie ein RS232-Modul bereit.

Die Datenübertragung beim DDS 10 erfolgt mit einer Datenübertragungsrate von 9600 Bit/s mit 8 Datenbits, 2 Stoppbits und gerader Parität. Nur mit diesen Einstellungen ist die Kommunikation möglich.

Der Funktionsgenerator DDS 110 hingegen verfügt bereits über einen USB-zu-UART-Schnittstellenwandler der Firma Silicon Laboratories auf der Platine. Durch diesen Baustein kann der Computer über die USB-Schnittstelle direkt mit dem DDS 110

67

Tabelle 1: Protokollrahmen des DDS 10		
Anzahl Bytes Beschreibung		
1	Startzeichen	
n	Nutzdaten	
1 Stoppzeichen		
Beispiel Protokollrahmen DDS 10:		

In diesem Beispiel wird an den DDS 10 der Befehl zum Setzen der Signalform Rechteck gesendet. 0×02 0×73 0×05 0×82 0×03 0×03 0×02 0×03 0×03 $0 \times$

kommunizieren, dazu ist nur der beigefügte Treiber zu installieren. Es wird keine zusätzliche Hardware benötigt.

Beispielprogramm

Startzeichen

Um den Einstieg zur Programmierung eines eigenen Software-Interfaces für den DDS 110 zu erleichtern, wird auf der Produktseite des DDS 110 der Quellcode eines kleinen Konsolenprogramms angeboten. Zusätzlich sind alle notwendigen Datenblätter und Dateien zur Verwendung des USB-zu-UART-Schnittstellenwandlers CP2102 von Silicon Laboratories zusammengefasst. Der DDS 110 überträgt seine Daten mit einer Datenrate von 76.800 Bit/s, 8 Datenbits, 1 Stoppbit und keiner Parität. Diese Einstellungen sind dann im eigenen Programmcode vorzunehmen.

Protokollrahmen

Kommen wir damit zur eigentlichen Protokollbeschreibung. Jeder gesendete Befehl und die dazugehörige Antwort wird in einem definierten Protokollrahmen übertragen. Innerhalb dieses Rahmens werden die eigentlichen Nutzdaten übertragen. Der Aufbau dieser Protokollrahmen ist in den Tabellen 1 und 2 dargestellt. Wie aus den Tabellen zu erkennen ist, sind die Protokollrahmen der beiden DDS-Boards unterschiedlich aufgebaut. Beim DDS 10 wird ein Datenpaket mit dem Startzeichen "STX" (0x02) eingeleitet und mit dem Stoppzeichen "ETX" (0x03) beendet. Die Zeichen stammen aus dem ASCII-Zeichensatz.

Diese Start- und Stoppzeichen dürfen innerhalb der Nutzdaten nicht wieder vorkommen, da das Datenpaket in diesem Fall nicht korrekt ausgewertet werden kann. Falls entsprechende Zeichen in den Nutzdaten vorhanden sind, müssen diese umcodiert werden. Dazu wird das ASCII-Steuerzeichen "ENQ" (0x05) verwendet und nach dem Schema in Abbildung 1

$\langle STX \rangle \rightarrow \langle ENQ \rangle \langle 0x82 \rangle$	
$\langle ETX \rangle \rightarrow \langle ENO \rangle \langle 0x83 \rangle$	
$\langle ENO \rangle \rightarrow \langle ENO \rangle \langle 0x85 \rangle$	

Bild 1: Codierungsanweisung von Steuerzeichen beim DDS 10

labelle 2: Prot	tokollra	hmen de	S DDS 110	
Anzahl Bytes	Besch	reibung		
1	Startzei	chen		
1	Paketnu	ımmer (is	t immer NULL)	
2	Länge d	ler Nutzda	aten	
n	Nutzdat	ten		
2	CRC16-	Checksun	nme	
In diesem Beispiel wird Setzen der Signalform 0x02 0x00 0x02 0 	d an den Rechtec <u>0x00</u>	DDS 110 k gesende <u>0x73</u> 	der Befehl zum et. 0×10 0×82	0xFE 0xF7 Checksumme
 Nutzdat	onlängo	 Befehl (codierter Parar (0x02 → Recht s')	neter teck)
Paketnummer	emanye			

umgesetzt. Ebenfalls ist in Beispielen zu den Protokollrahmen die Verwendung dieser Codierung zu sehen.

Startzeichen

Im Gegensatz zum DDS 10 ist der Protokollrahmen des DDS 110 etwas umfangreicher. Es werden zusätzliche Informationen übertragen, die der Kommunikationssicherheit dienen. Der Protokollrahmen des DDS 110 beginnt wie der des DDS 10 mit dem Startzeichen "STX". Anschließend wird eine Paketnummer übertragen, diese hat beim DDS 110 jedoch keine Relevanz und besitzt deshalb immer den Wert NULL. Die nachfolgenden 2 Byte enthalten die Länge der nicht codierten Nutzdaten. Dabei wird zuerst das untere und anschließend das obere Byte dieses 16-Bit-Wertes übertragen. Nun folgen die Nutzdaten selbst, auch hier erfolgt wieder eine Umcodierung der Daten. Anders als beim DDS 10 braucht das Stoppzeichen "ETX" nicht beachtet zu werden, da es der Protokollrahmen des DDS 110 nicht nutzt. Außerdem ist das zur Umcodierung genutzte ASCII-Steuerzeichen hier das "DLE" (0x10) anstatt des "ENQ". Abbildung 2 zeigt das Schema der Umcodierung beim DDS 110.

Nach den Nutzdaten folgt zum Abschluss die 2 Byte lange CRC16-Checksumme. Sie wird mit dem Polynom 0x8005 nach dem allgemein bekannten Berechnungsverfahren bestimmt. Die Checksumme wird über alle übertragenen Zeichen gebildet, für die beiden Checksummenbytes selbst wird jeweils eine NULL eingesetzt. Auch hier erfolgt zuerst die Übertragung des unteren und anschließend des oberen Bytes dieses 16-Bit-Wertes.

Befehlsübersicht

Die beiden DDS-Boards verfügen für die Kommunikation mit

 $\langle STX \rangle \rightarrow \langle DLE \rangle \langle 0x82 \rangle$ $\langle DLE \rangle \rightarrow \langle DLE \rangle \langle 0x90 \rangle$

Bild 2: Codierungsanweisung von Steuerzeichen beim DDS 110

Tabelle 3: Gemeinsame Befehle des DDS 10 und DDS 110			
Befehl	Hex-Code	Beschreibung	
х	0x78	PC-Verbindung de-/aktivieren	
V	0x56	Geräteversion auslesen	
f	0x66	Frequenz einstellen	
F	0x46	Frequenz auslesen	
S	0x73	Signalform einstellen	
S	0x53	Signalform auslesen	
a	0x61	Sinusfilter schalten	
A	0x41	Status des Sinusfilters auslesen	
t	0x74	PLL-Teilerfaktor einstellen	
Т	0x54	PLL-Teilerfaktor auslesen	
0	0x6F	PLL-Offset einstellen	
0	0x4F	PLL-Offset auslesen	
k	0x6B	Kalibrierwert des Oszillators einstellen	
K	0x4B	Kalibrierwert des Oszillators auslesen	
w	0x77	Wobble-Mode konfigurieren	
m	0x6D	Modulations-Mode konfigurieren	
е	0x65	Betriebsmodus einstellen	
E	0x45	Betriebsmodus auslesen	

einem Computer über einen Befehlssatz mit einer Vielzahl von Befehlen. Die folgenden Tabellen geben eine Übersicht hierzu. In der Tabelle 3 sind zunächst die von DDS 10 und DDS 110 gemeinsam benutzten Befehle dargestellt. Die beiden Tabellen 4 und 5 zeigen hingegen Befehle, die speziell zu einem der DDS-Boards gehören.

Beschreibung der Befehle

Nachdem wir uns eine generelle Übersicht über die verfügbaren Befehle verschafft haben, sollen diese in der Folge genauer erläutert werden. Dazu ist es notwendig, die Grundlagen der Kommunikation zwischen PC und DDS-Board zu verstehen. Neben dem Protokollrahmen sind auch die Nutzdaten von gesendeten Befehlen und deren Antworten klar strukturiert.

Aufbau eines Befehls

Im ersten Byte der Nutzdaten ist das Befehlswort enthalten, im Anschluss folgen die Parameter.

<Befehl><Parameter 1>...<Parameter n>

Tabelle 5: Spezielle Befehle des DDS 110		
Befehl	Hex-Code	Beschreibung
р	0x70	Aktuelle Konfiguration programmieren
Р	0x50	Aktuelle Konfiguration auslesen
С	0x43	Minimal-/Maximalfrequenz einstellen
Z	0x7A	Frequenzschrittweite einstellen
Z	0x5A	Frequenzschrittweite auslesen
g	0x67	Kalibrierte Amplitude einstellen
d	0x64	Amplitude digital einstellen
q	0x71	Amplitude kalibrieren
r	0x72	Reset auf Werkseinstellungen

Tabelle 4: Spezielle Befehle des DDS 10			
Befehl	Hex-Code	Beschreibung	
В	0x42	PC-Verbindung manuell beendet	
р	0x70	Speicherplatz programmieren	
Р	0x50	Speicherplatz auslesen	

Einige Befehle benötigen keine Parameterangabe, bei diesen Befehlen bestehen die Nutzdaten nur aus dem Befehlswort.

Aufbau einer Antwort

Nachdem ein Befehl zum DDS-Board gesendet wurde, erwartet der PC eine Antwort auf den Befehl. Das erste Byte der Nutzdaten enthält das zuvor gesendete Befehlswort, damit ist eindeutig zu erkennen, worauf geantwortet wird. Nach dem Befehlswort werden die Antwortparameter übertragen:

<Befehl><Antwort 1>...< Antwort n>

Neben den Antworten, die konkrete Werte wie die aktuelle Frequenz zurückgeben, gibt es auch allgemeine Antworten. Diese werden eingesetzt, um einfache Befehle zu bestätigen oder einen Fehler zu zeigen. Hierfür werden zwei ASCII-Steuerzeichen eingesetzt. Mit dem Zeichen "ACK" (0x06) werden positive Quittierungen bestätigt. Das Zeichen "NAK" (0x15) weist auf einen Fehler hin, dies kann durch einen unbekannten Befehl oder durch ungültige Parameter verursacht werden.

Bei Parametern und Antworten, die einen 16-Bit- bzw. 32-Bit-Wert benötigen, wird zuerst das oberste und dann die nachfolgenden Bytes übertragen, wie es im nachfolgenden Beispiel zu sehen ist.

Beispiel: Setzen einer Frequenz beim DDS 10: In diesem Beispiel wird an den DDS 10 der Befehl zum Setzen der Frequenz auf 50 kHz gesendet.				
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	<u>0x03</u>			
	Stopp-			
Parameter (50.000,0 Hz) zeichen				
Betehl (,t')				
Startzeichen				

Eine detaillierte Übersicht der Befehle und der zu erwartenden Antworten ist in der Befehlstabelle dargestellt, die Sie auf der Produktseite einsehen können.

Internet:

Herstellerseite des USB-zu-UART-Schnittstellenwandlers CP2102: http://www.silabs.com

Produktseite des DDS 10: http://www.elv.de/output/controller.aspx?cid=74&detail=10&detail2=10617

Produktseite des DDS 110: http://www.elv.de/output/ controller.aspx?cid=74&detail=10&detail2=16755