

Signalgenerator

Hendrik Lüth

18. Mai 2015

Inhaltsverzeichnis

1	Lastenheft Signalgenerator	5
1.1	Zielbestimmung	6
1.2	Funktionale Anforderungen	6
1.2.1	Platine & Layout	6
1.2.2	Beschaffen der Bauteile	6
1.2.3	Bestücken der Platine	6
1.2.4	Firmware des Mikrocontrollers	6
1.3	Anhang	7
1.3.1	Gehäuseplan	7
2	Lastenheft Gehäuse	9
2.1	Auftrag	10
2.2	Gehäuseplan	10
2.3	Änderungen am Gehäuse	11
3	Kostenübersicht	13
4	Inbetriebnahmeprotokoll	17
4.1	Optische Kontrolle	18
4.2	Elektrische Kontrolle	18
4.3	Funktionskontrolle	19
5	Arbeitsablaufplan	21
6	com-protokoll	23
6.1	Allgemeines	24
6.2	Aufbau einer Kommunikationseinheit	24
6.3	Aufbau einzelner Befehle	24
6.3.1	Computer → Signalgenerator	25
6.3.2	Signalgenerator → Computer	26
6.4	Berechnung der Registerwerte	27
6.4.1	Frequenz	27
6.4.2	Signalform	28
6.4.3	Spitzenspannung	28
6.4.4	Offsetspannung	29
6.4.5	Sonstige Register	29
6.5	Errorcodes	30

Kapitel 1

Lastenheft Signalgenerator

1.1 Zielbestimmung

Es soll ein Signalgenerator angefertigt werden. Hierbei ist sich an den Schalplan und die dort verzeichneten Werte zu halten. Die verwendeten Bauteile sollen, soweit möglich, SMD-Bauteile sein.

Der Umfang des Auftrages umfasst:

- das Layouten und Herstellen der Platine
- Beschaffung der Bauteile
- das Bestücken der Platine
- das programmieren einer Software für den Mikrocontroller des Signalgenerators

1.2 Funktionale Anforderungen

1.2.1 Platine & Layout

Da ein bestimmtes Gehäuse verwendet werden soll darf die Platine des Signalgenerators nicht größer als die Innenbemaßung des Gehäuses sein. Eine Technische Zeichnung des Gehäuses ist dem Lastenheft beigelegt.

Des weiteren sollte das Layout so angefertigt werden, dass die Mikro-USB Buchse und die SMA-R Buchse gegenüber von einander an den kurzen Seite der Platine platziert werden. Die Platine soll Doublelayer sein und eine Dicke von 1,6mm haben.

1.2.2 Beschaffen der Bauteile

Die Bauteile sollen so günstig wie möglich beschafft werden.

1.2.3 Bestücken der Platine

Die Platine ist vollständig zu bestücken und optisch auf Kurzschlüsse zu prüfen.

1.2.4 Firmware des Mikrocontrollers

Die Firmware für den Mikrocontroller soll in C geschrieben werden.

Zur Ansteuerung des USB-Controllers soll das LUFA Framework verwendet werden.

Der Mikrocontroller soll in der Lage sein Daten vom Computer über USB zu empfangen und diese an die entsprechende Peripherie weiterzugeben. Für die Peripherie sollen ebenfalls die in LUFA enthaltenen Bibliotheken benutzt werden.

Für die USB-Kommunikation soll der Mikrocontroller die VID 0x03EB und die PID 0x204F benutzen und sich als Vendorspezifisches HID Gerät anmelden. Für die Kommunikation sollen nur HID-Reports verwendet werden. Der Mikrocontroller soll in der Lage sein bestimmte Daten zurück an den PC zu geben, um Fehleranalyse und Fehlererkennung auf der PC-Seite durchführen zu können. Auch hierzu ist der Aufbau der zu übertragenden Daten in der Angelegten Protokollspezifikation zu finden.

Kapitel 2

Lastenheft Gehäuse

2.3 Änderungen am Gehäuse

Hier kommt die Technische Zeichnung rein...

Kapitel 3

Kostenübersicht

Kostenübersicht

Projekt: Signalgenerator

In der folgenden Tabelle sind alle Kosten für die Herstellung des Signalgenerators zusammengestellt. In den Kostenpunkten für die Herstellung des Signalgenerators und für den Bau des Gehäuses sind neben den Materialkosten auch die Lohnkosten enthalten. Eine genaue Aufschlüsselung hierzu ist in den Angeboten auf den folgenden Seiten zu finden.

Pos.	Posten	Menge	Einzelpreis	Gesamtpreis
1	Herstellung des Signalgenerators	1	528,79€	528,79€
2	Herstellung des Gehäuses des Signalgenerators	1	30,00€	30,00€
3	Montagematerial für Montage des Signalgenerators	1	10,00€	10,00€
4	Arbeitszeit	20	30,00€	600,00€
			MwSt.:	222,07€
Alle Preise inkl. 19% Gesetzl.MwSt.			Gesamt:	1168,79€

Flensburg, 04.05.2015

Werkstatt 2
Ausbildungswerkstatt Flensburg
Mürwiker Str. 203
24944 Flensburg

Hendrik Lüth
Ausbildungswerkstatt Flensburg
Mürwiker Str. 203
24944 Flensburg

Angebot für die Herstellung des Signalgenerators

Sehr geehrter Herr Lüth,

ich übersende ihnen das Angebot zur Herstellung des Signalgenerators nach Ihrem Schaltplan und Wünschen.

Pos.	Posten	Menge/Zeit	Einzelpreis	Gesamtpreis
1	Anfertigung des Platinenlayouts	10 Std.	30,00€	300,00€
2	Herstellung der Platine (externer Auftrag)	1	14,84€	14,84€
3	Bauteilkosten	1	53,95€	53,95€
4	Beschaffung der Bauteile	2 Std.	20,00€	40,00€
5	Bestücken der Platine	3 Std.	35,00€	105,00€
6	Prüfen des Signalgenerators	0,5 Std.	30,00€	15,00€
			MwSt.:	100,47€
Alle Preise inkl. 19% Gesetzl.MwSt.			Gesamt:	528,79€

Bitte melden Sie sich bei uns, wenn ihnen das Angebot zusagt, wir werden dann mit der Herstellung beginnen. Das Angebot besitzt eine Gültigkeit von 2 Wochen.

mit freundlichen Grüßen,

Max Mustermann

Flensburg, 04.05.2015

Werkstatt 1
 Ausbildungswerkstatt Flensburg
 Mürwiker Str. 203
 24944 Flensburg

Hendrik Lüth
 Ausbildungswerkstatt Flensburg
 Mürwiker Str. 203
 24944 Flensburg

Angebot für die Herstellung eines Gehäuses

Sehr geehrter Herr Lüth,

ich übersende ihnen das Angebot zur Herstellung zur Herstellung eines Gehäuses.
 Aufgrund des niedrigen Umfangs des Projektes ist es uns möglich Ihnen eine
 Pauschalpreis von 30,00€ anbieten zu können. Dieser beinhaltet Beschaffung und
 Anfertigung des Gehäuses.

Pos.	Posten	Menge/Zeit	Einzelpreis	Gesamtpreis
1	Materialkosten Gehäuse	1	10,00€	10,00€
2	Herstellung des Gehäuses	1	20,00€	20,00€
			MwSt.:	5,70€
Alle Preise inkl. 19% Gesetzl.MwSt.			Gesamt:	30,00€

Bitte melden Sie sich bei uns, wenn ihnen das Angebot zusagt, wir werden dann mit der
 Herstellung beginnen. Das Angebot besitzt eine Gültigkeit von 2 Wochen.

mit freundlichen Grüßen,

Max Mustermann

Kapitel 4

Inbetriebnahmeprotokoll

Inbetriebnahme-Protokoll Signalgenerator

Name des Prüfers:	
Prüfdatum:	
Seriennummer:	

4.1 Optische Kontrolle

Nr.	Prüfauftrag	Ja	Nein
1	Sind alle Bauteile bestückt?		
2	Sind alle Bauteile ordnungsgemäß befestigt?		
3	Sind IC-Beine miteinander verbunden, die nicht miteinander verbunden sein dürfen?		
4	der Fädeldraht zum aktivieren der USB-Schnittstelle des Mikrocontrollers ist eingelötet		
5	Der Lötjumper zum aktivieren des LT1615 ist gesetzt		

4.2 Elektrische Kontrolle

Die folgenden Messungen sind mit einem Digitalmultimeter durchzuführen. Sollte einer der Widerstände nicht den Anforderungen entsprechen, so darf die Platine unter keinen Umständen einer Funktionskontrolle unterzogen werden.

Nr.	Prüfauftrag	Ja	Nein	Wert
1	Messen des Widerstandes zwischen V_{cc} und GND. Ist der Wert größer als 900Ω ?			
2	Messen des Widerstandes zwischen USB_{D+} und GND. Ist der Wert größer als $500k\Omega$?			
3	Messen des Widerstandes zwischen USB_{D-} und GND. Ist der Wert größer als $500k\Omega$?			
4	Messen des Widerstandes zwischen $+12V$ und GND. Ist der Wert größer als $50k\Omega$?			
5	Messen des Widerstandes zwischen $-12V$ und GND. Ist der Wert größer als $10k\Omega$?			
6	Messen des Widerstandes zwischen $+3.3V$ und GND. Ist der Wert größer als $2k\Omega$?			
7	Messen des Widerstandes zwischen $-3.3V$ und GND. Ist der Wert größer als $2k\Omega$?			

4.3 Funktionskontrolle

Sollte einer der Widerstände in der elektrischen Kontrolle nicht den Anforderungen entsprechen, so darf die Platine unter keinen Umständen einer Funktionskontrolle unterzogen werden.

Starten Sie das Testprogramm am Computer und stecken Sie den Signalgenerator an.

Nr.	Prüfauftrag	Ja	Nein
1	Wird der Signalgenerator vom Computer erkannt?		
2	Starten sie den Software-Test. Ist der Test erfolgreich verlaufen?		
3	Messen sie den Master-Clock (MCLK) an Pin 5 des AD9833. Ist der Wert im 5% Rahmen von 25MHz?		

Tragen Sie die Frequenz des Master-Clock in das entsprechende Feld für Kalibrierungswerte ein. Starten Sie die Messung Nr.1 bis Nr.4 und folgen Sie den Anweisungen in der Software. Die enthalten auch die Einstellungen für das Oszilloskop. Vergleichen Sie das Bild auf dem Oszilloskop dem Bild der Beispielmessung. Sind die Bilder annähernd identisch? Toleranzen im Bereich von $\pm 5\%$ sind akzeptabel.

Nr.	Prüfauftrag	Ja	Nein
4	Messung Nr. 1		
5	Messung Nr. 2		
6	Messung Nr. 3		
7	Messung Nr. 4		
8	Messen sie die Ausgangsspannung an Pin 10 des AD9833. Liegt ihr Wert bei 700mV ± 100 mV?		

Tragen Sie die Ausgangsspannung in das entsprechende Feld für Kalibrierungswerte ein.

Nr.	Prüfauftrag	Ja	Nein
9	Führen Sie den Speichertest durch. War der Test erfolgreich?		
10	Führen Sie den Lesetest durch. War der Test erfolgreich?		

Wenn alle Tests erfolgreich waren tragen Sie die Seriennummer des Gerätes in das entsprechende Feld ein und schreiben Sie die Kalibrierungswerte auf den Signalgenerator.

Trennen Sie den Signalgenerator Ordnungsgemäß von der Software und vom Computer.

Kapitel 5

Arbeitsablaufplan

Arbeitsablaufplan Signalgenerator

Projektleiter: Hendrik Lüth

Projektzeitraum: 01.05.2015 - 31.05.2015

Pos.	Arbeitsschritt	Soll	
1	Anfertigung eines Arbeitsablaufplans	1,0h	
2	Schreiben eines Lastenheftes für die Herstellung des Signalgenerators	2,5h	
3	Anfertigung eines Inbetriebnahmeprotokolls für den Signalgenerator	2,5h	
4	Schreiben eines Lastenheftes für die Herstellung des Gehäuses	2,0h	
5	Anfertigung eines Abnahmeprotokolls für das Gehäuse	0,5h	
6	Ausfüllen des Inbetriebnahmeprotokolls für den Signalgenerator	1,0h	
7	Ausfüllen des Abnahmeprotokolls des Gehäuses	1,0h	
8	Montage des Signalgenerators in das Gehäuse	0,5h	
9	Aufstellen einer Kostenübersicht des Projektes	3,0h	
		13,0h	

Kapitel 6

com-protokoll

6.1 Allgemeines

In diesem Dokument wird die Datenübertragung zwischen dem Mikrocontroller des Signalgenerators und eines Computers definiert. Die Daten werden über den USB-Bus übertragen. Die USB-Spezifikationen¹ enthalten alle nötigen Informationen, welche für Kommunikationen über den Bus nötig sind.

Der Signalgenerator wird als HID (Human Interface Device) am Computer angemeldet, wodurch keine Installation von zusätzlichen Treibern nötig ist. Die Übertragung der Daten erfolgt über HID-Reports. Zum aktuellen Zeitpunkt benutzt LabConnect für den Signalgenerator die VID 0x1209 und die PID 0x2222, welche unter Linux als GenericHID-Gerät von InterBiometrics zu finden ist. Da es sich bei der VID um eine VID handelt, welche für OpenSource Projekte gedacht ist, ist es fraglich ob der Signalgenerator je richtig angezeigt wird. Von dem Kauf einer eigenen VID für LabConnect wird derzeit abgesehen.

6.2 Aufbau einer Kommunikationseinheit

Eine Kommunikationseinheit, im folgenden als Paket bezeichnet, besteht aus 13 Byte. Jedes Paket hat einen 1 Byte großen Header an seinem Anfang und einen 1 Byte großen Tail an seinem Ende. Der Header enthält die Paket-ID, an welcher sich Flussrichtung der Daten und Art der Daten erkennen lassen. Ist das 5. Bit des Headers gleich 0, so ist die Flussrichtung der Daten vom Computer zum Mikrocontroller, ist es gleich 1 vom Mikrocontroller zum Computer. An den unteren 4 Bit lässt sich der Typ des Paketes erkennen. In der folgenden Tabelle sind alle Befehle nach Paket-ID sortiert aufgelistet:

Paket-ID	Flussrichtung	Bezeichnung	Größe der Daten
0x00	PC → μ C	Config-Request	1 Byte
0x01	PC → μ C	Set-Command	12 Byte
0x02	PC → μ C	Data-Request	0 Byte
0x03	PC → μ C	Error/Status-Request	0 Byte
0x10	μ C → PC	Config-Response	10 Byte
0x12	μ C → PC	Data-Response	12 Byte
0x13	μ C → PC	Error/Status-Response	5 Byte

6.3 Aufbau einzelner Befehle

In diesem Abschnitt wird der Aufbau einzelner Befehle erläutert. Ob ein Befehl vom Computer zum Signalgenerator geht ist an der Paket-ID zu erkennen. Dies ist im Abschnitt „Aufbau einer Kommunikationseinheit“ beschrieben.

¹http://www.usb.org/developers/docs/usb20_docs/usb_20_031815.zip

6.3.1 Computer → Signalgenerator

Config-Request

Der Config-Request steht am Anfang jeglicher Kommunikation zwischen Signalgenerator und Computer nach dem anstecken des Signalgenerators. Der Config-Request fragt beim Signalgenerator diverse Kalibrierungsdaten wie die Frequenz des Referenztaktes oder die Boot-Daten an.

Byte	Wert	Beschreibung
0	0x00	Paket-ID
1	0x55	Prüfdaten, damit der Inhalt des Paketes nicht null ist. Der Wert ist auf 0x55 festgesetzt.

Set-Command

Mit dem Set-Command werden alle nötigen Informationen wie Frequenz, Registerwerte für die digitalen Potentiometer und Bootdaten übergeben. Die Folgende Tabelle zeigt den Aufbau:

Byte	Wert	Beschreibung
0	0x01	Paket-ID
1	*	Diese beiden Bytes enthalten die Daten für das Kontrollregister des AD9833.
2	*	
3	*	Diese vier Bytes enthalten die Daten für das Frequenzregister des AD9833. Die Berechnung dieser Werte ist im Verlauf dieses Dokumentes erklärt.
4	*	
5	*	
6	*	
7	*	In diesen beiden Bytes sind die Registerwerte des Digi-Poti für die Ausgangsspannung enthalten.
8	*	
9	*	In diesen beiden Bytes sind die Registerwerte des Digi-Poti für die Offset-Spg. enthalten.
10	*	
11	*	Multiplexer
12	*	Bootdaten

Data-Request

Nach einem Config-Request werden die Daten ausgewertet. Sollten die Bootdaten anzeigen, dass bereits beim einschalten des Signalgenerators die gespeicherte Konfiguration geladen wurde, so wird ein Data-Request gesendet, um herauszufinden wie die Konfiguration ist um sie später in der graphischen Oberfläche anzuzeigen. Dieses Paket hat keine Nutzdaten.

Byte	Wert	Beschreibung
0	0x02	Paket-ID

Error/Status-Request

Ein Error/Status-Request kann zu jedem Zeitpunkt, z.B. nach einer Datenübertragung gestellt werden um den Status des Systems zu prüfen. Dieses Paket enthält keine Nutzdaten.

Byte	Wert	Beschreibung
0	0x03	Paket-ID

6.3.2 Signalgenerator → Computer

Config-Response

Byte	Wert	Beschreibung
0	0x10	Paket-ID
1	*	Seriennummer
2	*	Bootdaten
3	*	Kalibrierungs-Daten des DDS-IC Frequenz des MCLK in Hz
4	*	
5	*	
6	*	
7	*	Kalibrierungs-Daten für das Digi-Poti Multiplikatoren für Berechnung
8	*	
9	*	Wert der Ausgangsspannung in mV_{ss}
10	*	

Data-Response

Dieses Paket ist die Antwort auf einen Data-Request. Es enthält die selben Daten wie ein Set-Command. Die Daten müssen vom Host dann in Frequenzen und Spannungen umgerechnet werden.

Byte	Wert	Beschreibung
0	0x01	Paket-ID
1	*	Diese beiden Bytes enthalten die Daten für das Kontrollregister des AD9833.
2	*	
3	*	Diese vier Byte enthalten die Daten für das Frequenzregister des AD9833. Die Berechnung dieser Werte ist im Verlauf dieses Dokumentes erklärt.
4	*	
5	*	
6	*	
7	*	In diesen beiden Bytes sind die Registerwerte des Digi-Poti für die Ausgangsspannung enthalten.
8	*	
9	*	In diesen beiden Bytes sind die Registerwerte des Digi-Poti für die Offset-Spg. enthalten.
10	*	
11	*	Multiplexer
12	*	Bootdaten

Error/Status-Response

Der Error/Status-Response enthält alle Error/Status-Meldungen die angefallen sind.

Byte	Wert	Beschreibung
0	0x13	Paket-ID
1	*	Error-Codes, bis zu 5 Stück.
2	*	
3	*	
4	*	
5	*	

6.4 Berechnung der Registerwerte

In dieser Sektion ist aufgelistet, wie die Registerwerte für den Signalgenerator berechnet werden. Es ist sich zwingend an die Formeln zu halten, da der Signalgenerator sonst nicht die gewünschten Ausgangssignale liefert.

6.4.1 Frequenz

Die Frequenzregister sind die Register, welche die Frequenz des Ausgangssignals kontrollieren. Mit einer Formel muss in Abhängigkeit vom Referenztakt und der gewünschten Frequenz des Ausgangssignals der Wert für dieses Register errechnet werden. Hier die allgemeine Formel:

$$\text{Registerwert} = F_{out} \div \frac{F_{MCL}}{2^{28}}$$

Und hier ein Beispiel für die Werte $F_{MCLK} = 25MHz$ und $F_{out} = 7,325MHz$:

$$\begin{aligned} \text{Registerwert} &= F_{out} \div \frac{F_{MCL}}{2^{28}} = 7,325MHz \div \frac{25MHz}{2^{28}} \\ \text{Registerwert} &= 78651588,61 \approx 78651589 \end{aligned}$$

In diesem Fall ist es möglich zu runden, da ein Bit nur c.a. 0,093Hz entsprechen. Nun muss der Wert noch in Binär umgerechnet werden:

$$\text{Dec} "78651589" = \text{Bin} "100\ 1011\ 0000\ 0010\ 0000\ 1100\ 0101"$$

Um den Wert in das Frequenzregister zu schreiben wird der binäre Wert in LSBs und MSBs aufgeteilt und hängen die Adressierung des Registers "01" und die fehlenden Nullen, um auf 28Bit zu kommen, davor:

MSBs: 0101 0010 1100 0000

LSBs: 0110 0000 1100 0101

Dies ist ein Beispielcode in C++, in dem die entsprechenden Register berechnet werden:

```

1 float mclk = 25000000, register_size = 268435456;
2 float teiler = mclk / register_size;
3 int f_regwert = frequenz / teiler;
4 //block1=lsb Block4=msb
5 unsigned char block1, block2, block3, block4;
6
7 block1 = f_regwert;
8 f_regwert = f_regwert >> 8;
9 block2 = f_regwert;
10 block2 = (block2 | 0x40) & (~0x80);
11 f_regwert = f_regwert >> 6;
12 block3 = f_regwert;
13 f_regwert = f_regwert >> 8;
14 block4 = f_regwert;
15 block4 = (block4 | 0x40) & (~0x80);

```

6.4.2 Signalform

Für das Register der Signalform gibt es nicht viel zu berechnen, da es nur drei Signalformen gibt. Die 2 Byte, mit denen die Signalform gesteuert wird können folgende Werte annehmen:

Wert	Signalform
0x2000	Sinus
0x2002	Dreieck
0x0000	Rechteck

Es ist hierbei darauf zu achten, dass auch der Zustand des Multiplexers angepasst wird, da es ansonsten zu unerwünschten Ausgangsspannungen kommen kann.

6.4.3 Spitzenspannung

Die Amplitude des Ausgangssignals lässt sich über den Multiplexer und das Digitalpotentiometer einstellen. Hierzu wird die Ausgangsspannung des DDS-IC als Berechnungsgrundlage hinzu gezogen. Mit dem Multiplexer kann man auswählen, ob das Signal direkt auf den Verstärker gegeben wird oder ob eine Teilung von 5:1 bzw eine Verstärkung von ungefähr 3 stattfinden soll, bevor das Signal auf den Verstärker gegeben wird. Welchen Wert das entsprechende Byte annehmen muss ist unter S5 sonstige Register im Unterabschnitt Multiplexernachzulesen. Die Registerwerte des Digitalpotentiometers werden wie folgt berechnet:

$$Registerwert_{gesamt} = \left(\frac{100k\Omega}{\frac{U_{Ausgang}}{U_{Eingang}} - 1} - 2,2k\Omega \right) \div \frac{200k\Omega}{512}$$

Hier ein Beispiel für die Werte $U_{Ausgang} = 7,5V_{ss}$ und $U_{Eingang} = 1V_{ss}$:

$$Registerwert_{gesamt} = \left(\frac{100k\Omega}{\frac{7,5V_{ss}}{1V_{ss}} - 1} - 2,2k\Omega \right) \div \frac{200k\Omega}{512}$$

$$Registerwert_{gesamt} \approx 13184,61 \div 390,625$$

$$Registerwert_{gesamt} \approx 34$$

Da der Registerwert für zwei in Reihe geschaltete Widerstände gilt muss dieser Wert noch auf beide Register aufgeteilt werden. Sollte das Ergebnis eine ungerade Bit-Zahl annehmen so erhält eines der Register einfach ein Bit mehr. Daraus ergibt sich, dass beide Register den dezimalen Werte "17" haben bzw 0x11 in Hexadezimal.

Der Folgende Beispielcode ist in C# geschrieben und berechnet den Gesamtwert beider Register.

```

1 int umax = 12000, bits = 512, register1, register2;
2 int ergebnis = u_amplitude_mv / (umax / bits);
3
4 if (510 < ergebnis)
5 {
6     ergebnis = 510;
7 }
8
9 if (ergebnis%2==0)
10 {
11     register1 = 255 - ergebnis / 2;
12     register2 = 255 - ergebnis / 2;
13 }
14 else
15 {
16     ergebnis = ergebnis - 1;
17     register1 = 255 - ((ergebnis / 2) + 1);
18     register2 = 255 - ergebnis / 2;
19 }

```

6.4.4 Offsetspannung

6.4.5 Sonstige Register

Bootdaten

Bootdaten	Beim Boot laden	Werte Speichern
0x00	Nein	Nein
0x01	Nein	Ja
0x10	Ja	Nein
0x11	Ja	Ja

Multiplexer**6.5 Errorcodes**

Fehlercode	Beschreibung des Fehlers
0x00	Kein Fehler
0x01	Keine gültige Package-ID
0x02	Transportdaten des Config-Requests sind falsch
0x03	Digitalpotentiometer ist nicht erreichbar
0x04	
0x05	
0x06	
0x07	