

VC0706 (UART/SPI) カメラモジュール

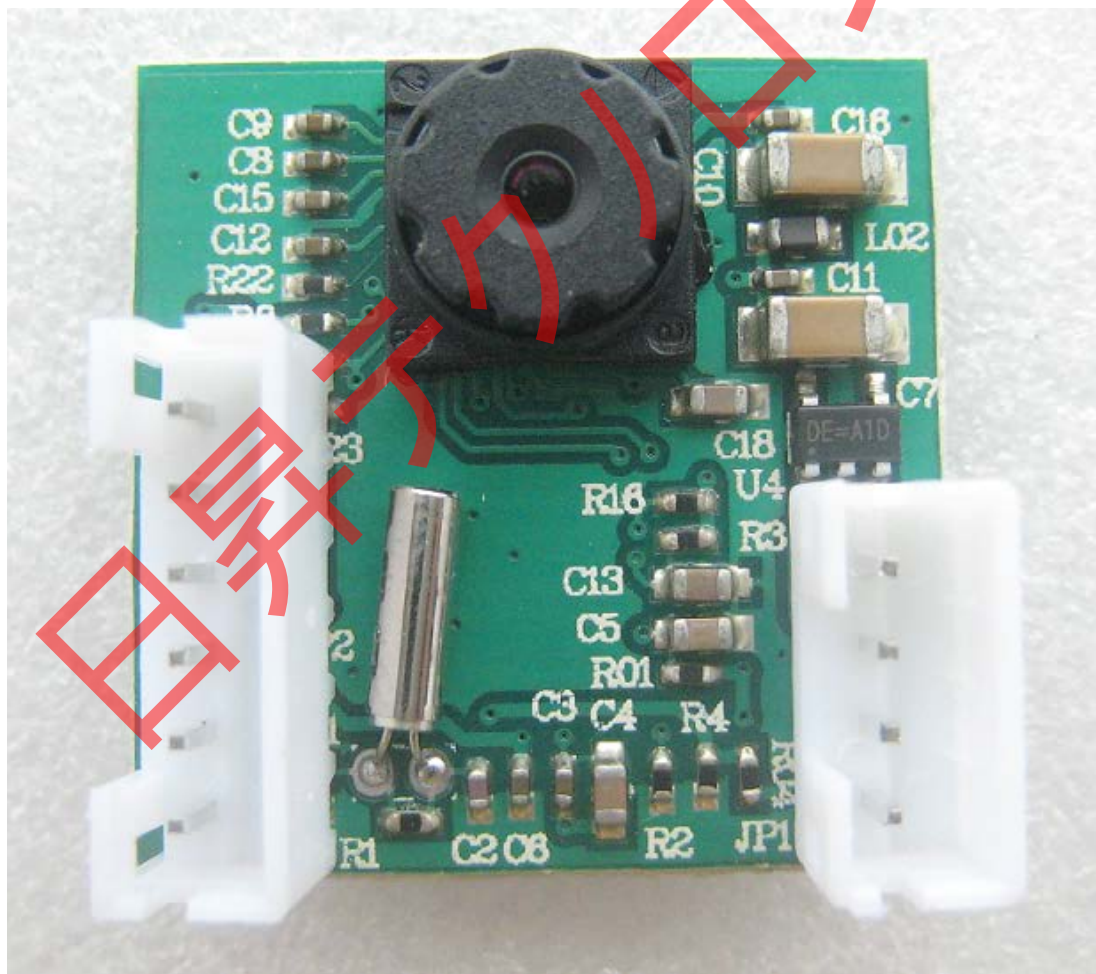
SPI 通信プロトコル

株式会社日昇テクノロジー

<http://www.csun.co.jp>

info@csun.co.jp

更新日 2014/10/27



copyright@2014

・修正履歴

NO	バージョン	修正内容	修正日
1	Ver1.0	新規作成	2014/10/27

※ この文書の情報は、文書を改善するため、事前の通知なく変更されることがあります。
最新版は弊社ホームページからご参照ください。「<http://www.csun.co.jp>」

※ (株)日昇テクノロジーの書面による許可のない複製は、いかなる形態においても厳重に
禁じられています。

目次

1、概要	4
2、SPI インターフェース	5
2.1. SPI Pin	5
2.2. SPI 動作モード	5
2.3. SPI 設定	5
2.3.1. Char Length	6
2.3.2. MSB first or LSB first	6
2.3.3. Polarity&Phase	6
2.3.4. 伝送スピード	6
2.4. 通信プロトコル	7
2.5. 通信コマンド	8
2.5.1. コマンドリスト	8
2.5.2. Frame Buffer	9
2.5.3. 操作手順	9
2.5.4. コマンドの詳細説明	10

1、概要

本文書はVC0706のSPI (Serial Peripheral Interface) の操作についての説明です。ユーザーが設計中にSPIでVC0706と通信して映像採集、または画像を書込んで、VC0706のJPEGデコーダーでデコードし、VC0706内蔵のVideo DACで出力したりする場合、ご参照ください。

VC0706チップの基本説明についてVC0706 Datasheetをご参照ください。

本文書はデバッグツールを動作させる外部マスターMCUを発信側、VC0706を受信側としている。

株式会社日昇テクノロジー

2、SPI インターフェース

2.1. SPI Pin

PIN NUMBER	NAME	TYPE	DESCRIPTION
80	UART_TXD	O	High speed UART data output
	GPIO14	O	GPIO 14 , only can be used as output mode
	SPI_MOD	I, PD	Hardware strap pin : SPI_MOD
89	SPI_SS	I/O, PU	SPI master interface slave select output
	GPIO10	I/O, PU	GPIO 10
90	SPI_SCK	I/O, PD	SPI master interface serial clock output
	GPIO11	I/O, PD	GPIO 11
91	SPI_MOSI	I/O, PD	SPI master interface master output
	GPIO12	I/O, PD	GPIO 12
92	SPI_MISO	I/O, PD	SPI master interface slave input
	GPIO13	I/O, PD	GPIO 13

2.2. SPI 動作モード

VC0706のSPIインターフェースは下記の2つのモードをサポートします。

- Master mode: VC0706はSPIのメインデバイス、主にはSPI Flashまたは他のスレーブSPIデバイスと繋ぐ場合使います。
- Slave mode: VC0706はSPIのスレーブデバイス、外部コントローラと繋ぐ場合使います。

VC0706のSPIインターフェースモードはチップリセットの間、ピン80（UART_TXD/GPIO14/SPI_MOD）のステータスによって選択されます。

- 0:VC0706はSPIのMaster modeで動作
- 1:VC0706はSPIのSlave modeで動作

外部マスターMCUがSPIでVC0706と繋いで画像採集、または画像を書込むコントロールする場合、VC0706はSPIのSlave modeで動作します。

外部マスターMCUがSPIでVC0706と繋いで画像採集、または画像を書込むコントロール場合、VC0706のSPIはSPI FlashまたはほかのSPIデバイスとの繋ぐ事はできない。

2.3. SPI 設定

外部マスターMCUがSPIでVC0706と繋ぐには、上記のピン80がチップリセット間のステータスによってSlave

modeを選択するほか、SPIの一部パラメータに対する設定が必要です。

2.3.1. Char Length

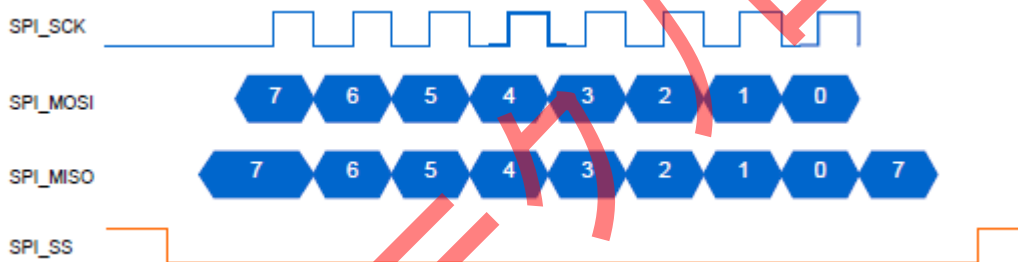
VC0706は8-bitのChar Lengthのみをサポートします。

2.3.2. MSB first or LSB first

VC0706はMSB firstの伝送方式のみをサポートします。

2.3.3. Polarity&Phase

SPIデータの有効時刻SPI-SCKの極性と上昇/下降状態によって、SPIが異なるpolarity&phaseをサポートします。VC0706のSPIは下記のモードのみをサポートします。

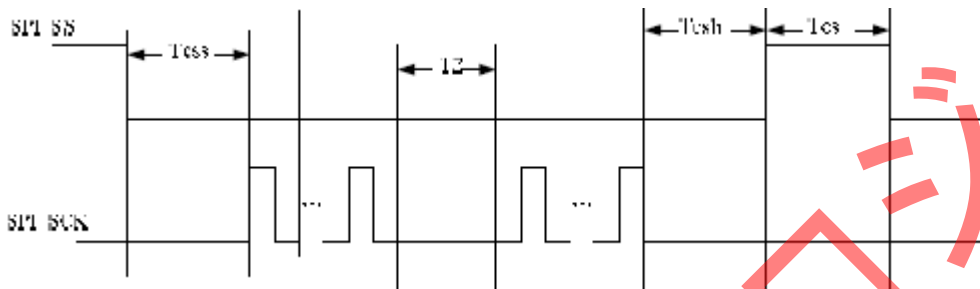


2.3.4. 伝送スピード

VC0706のSPIは異なるスピードをサポートしますので、ユーザーはEEPROMのアドレス0x000Eの変更で設定できます。

アドレス	機能	サイズ [バイト]	デフォルト値	機能説明
0x000E	SPI BPS	0x04	0x000D, 0x0200	SPIのボーレート 前の2つのバイトはSPIのDIVIDERレジスターの設定に使う、 後の2つのバイトはSPIのSSレジスターの設定に使う。 計算式： $\text{DIVIDER} = \text{SPI_CLK} / (2 * \text{BAUD}) - 1$ その中：SPI_MCLKは27Mに設定されている BAUDはSPIのボーレート DIVIDERはDIVIDERのレジスターに書き込む数値 SSレジスターに書き込むデフォルト値は0x0200である

よく使うボーレートのINTERとFRACの対照リスト：			
ボーレート	DIVIDER	SS	
0.844M	0x000F	0x0200	
1.688M	0x0007	0x0200	
3.375M	0x0003	0x0200	
6.750M	0x0001	0x0200	
13.500M	0x0000	0x0200	



Tcss: SPI_SS setup time(>4*Tspi_mclk)

Tcsh: SPI_SS hold time(>2*Tspi_mclk)

Tcs: SPI_SS high time(>2*spi_mclk)

Tspi_sck: SPI_SCKの周期

T2: 外部マスターMCUがSPIでデータブロックを送信する時間間隔

N1: 1つのフレームデータ送信に必要なの間隔数 (1フレームの送信に必要なデータブロック数-1)

N2: 1フレームで送信するデータ数

T: 1フレームの伝送に必要な総時間

$T = \text{TSPI_SCK} * 8 * N2 + N1 * T2 + Tcss + Tcsh + Tcs$

Tcss+Tcsh+Tcs: 基本的には無視してもいい

サイズが64kBの1フレームの転送を例として、SPI_MCLK=27MHz、最高伝送スピード (13.5M) で伝送する場合、外部マスターMCUのSPI Buffer=16バイトで、毎回SPI Bufferの処理タイムは5usにすれば、1フレームを伝送するには下記の時間が必要です：

$$T = (1/13.5) * 8 * (64 * 1024) + ((64 * 1024) / 16 - 1) * 5 + (4 + 2 + 2) * (1/27) = 59311 \mu s$$

2.4. 通信プロトコル

外部マスターMCUがSPIでVC0706と通信する通信プロトコルとVC0706のUARTシリアル通信プロトコルのフォーマットが完全に互換できます。

シリアル通信プロトコルのフォーマットは下記のとおりです。

受信のコマンドフォーマット：プロトコルフラグ(1バイト)+シリアル番号(1バイト)+コマンド字(1バイト)+データサイズ(1バイト)+データ(0~16バイト)

リプライのコマンドフォーマット：プロトコルフラグ(1バイト)+シリアル番号(1バイト)+コマンド字(1バイト)+ステータスバイト(1バイト)+データサイズ(1バイト)+データ(0~16バイト)；

その中：

プロトコルフラグ：このプロトコルをVC0706シリアルの通信プロトコルと示すのに使う；

受信のプロトコルフラグは0x56(‘V’)で、リプライのプロトコルフラグは0x76(‘v’)である

シリアル番号：複数のデバイスが同時につながる場合、それらのデバイスを区別するのに使う。シリアルコマンドのシリアル番号がデバイスのシリアル番号が一致する場合、デバイスが該当コマンドを受信できる。シリアル番号値の範囲は0~225である。

コマンド字：具体的なシリアル通信コマンドを示すのに使う

データのサイズ：後のデータのサイズを表示する。プロトコルフラグ、シリアル番号、コマンド字とデータサイズを除く。その値の範囲は0~16である。

データ：コマンドが使用するデータである。異なるコマンドがデータのサイズとフォーマットが異なり、最大16バイトである。

ステータスバイト：コントロールコマンド、データ書き込みコマンドに対して、データ先頭の1バイトはコマンドが成功したかどうかを示す。0なら成功、0x01~0xFFなら失敗を示す。

ステータス分類の意味は下記のリストをご覧ください：

ステータスコード	エラー説明
0	コマンドの実施が正しい
1	違反のコマンド字で、システムがコマンド字をサポートしないことを示す
2	違反のデータサイズで、このコマンドがこのサイズのデータサイズの数値をサポートしないことを示す
3	データフォーマット違反
4	現在のシステムステータスでコマンドの実施できない
5	コマンド受信したが、実施が失敗

説明：

- ・複数バイト数のデータの場合、データのオート方式は上位データが前に、下位データが後です。
- ・シリアル番号が間違いの場合、システムからエラー警告が出しません。
- ・コマンドのデータのサイズは最大16バイトまでです。
- ・コマンドのフォーマットまたはコマンドが実施失敗する場合、リプライのフォーマットはデフォルトで1バイトのエラーコードで、0バイトのデータサイズです。

2.5. 通信コマンド

2.5.1. コマンドリスト

コマンド定義	コマンド字	コマンドの意味
GET_VERSION	0x11	Firmwareプログラムバージョンを取得

SET_SERIAL_NUMBER	0x21	シリアル番号を設定
SET_PORT	0x24	シリアルプロパティを設定
SYSTEM_RESET	0x26	システムリセット
READ_DATA	0x30	デバイスのレジスターを読み出す
WRITE_DATA	0x31	デバイスのレジスターにデータを書き込む
READ_FBUF	0x32	FBUFから画像を読み出す
WRITE_FBUF	0x33	FBUFに画像を書き込む
GET_FBUF_LEN	0x34	FBUFの画像のサイズを取得
SET_FBUF_LEN	0x35	FBUFの画像のサイズを設定
FBUF_CTRL	0x36	FBUFをコントロール
COMM_MOTION_CTRL	0x37	シリアルMotionの監視を起動または停止
COMM_MOTION_STATUS	0x38	シリアルMotionの監視ステータスを取得
COMM_MOTION_DETECTED	0x39	シリアルMotion監視を起動する場合、シリアルがMotion検出を発信するのに使う
MIRROR_CTRL	0x3A	画像がMirror表示するかを設定
MIRROR_STATUS	0x3B	画像がMirror表示するかどうかの状態を取得
COLOR_CTRL	0x3C	カラーコントロール
COLOR_STATUS	0x3D	現在のカラーとカラーコントロールのモードを取得
POWER_SAVE_CTRL	0x3E	省エネモードに切り替えるかどうかをコントロール
POWER_SAVE_STATUS	0x3F	省エネモードに切り替えるかどうかを検査
AE_CTRL	0x40	AEの関連プロパティを設定
AE_STATUS	0x41	AEの関連プロパティを取得
MOTION_CTRL	0x42	Motionコントロール
MOTION_STATUS	0x43	Motionステータスを取得
TV_OUT_CTRL	0x44	ビデオ出力の起動と停止をコントロール
OSD_ADD_CHAR	0x45	OSDキャラクタを追加
GET_FLASH_SIZE	0x60	SPI Flashのサイズを取得
ERASE_FLASH_SECTOR	0x61	Flashの1ブロックをイレーズ
ERASE_FLASH_ALL	0x62	Flash全体をイレーズ
READ_LOGO	0x70	LOGOを読み出して表示
SET_BIZTMAP	0x71	Bitmapを操作
BATCH_WRITE	0x80	データをバッチで書く

2.5.2. Frame Buffer

このドキュメントでは、FBUFはFrame Bufferの意味です。VC0706内部は2つのFrame Bufferをサポートします。カレントフレームと次のフレームと呼ばれています。

2.5.3. 操作手順

外部マスターMCUがSPIでVC0706が採集したビデオを読み出す、またはSPIで画像をVC0706のFBUFに書き込むには操作手順を守る必要があります。

2.5.3.1. VC0706 から画像を読み出す

- FBUF_CTRLコマンドを発信して、パラメータは0x00に設定し、カレントフレームの更新を停止する
- GET_FBUF_LENコマンドを発信でカレントフレームの画像のサイズを取得する
- READ_FBUFコマンドを発信で画像データを読み取る。READ_FBUFコマンドのパラメータを下記のように設定する：
 - * FBUFフレームタイプを0x00に設定
 - * 操作モードを0x0Fに設定
 - * 開始アドレスを0x00に設定
 - * データサイズをGET_FBUF_LENコマンドで取得したデータサイズを設定
 - * 延長時間はデータとコマンドリプライの間に延長を追加して、データをコマンドリプライと区別するのに使う、デフォルト値は3000で、変更できる
- READ_FBUFコマンドを発信した後、VC0706のリプライを待ち、リプライが正しければ、VC0706から送信されるデータの取得を待つ。データ取得後、VC0706が再度コマンドリプライを発信して外部マスターMCUにデータ伝送完了の通知を出す
- 読み取り完了後、FBUF_CTRLコマンドを発信し、パラメータは0x02で、フレームの更新を復帰させる

2.5.3.2. VC0706 に画像を書き込む

- FBUF_CTRLコマンドを発信して、パラメータは0x02に設定し、カレントフレームの更新を停止する
- WRITE_FBUFコマンドを発信して画像データを読み出し、WRITE_FBUFコマンドのパラメータを下記のように設定する
 - * 操作モードを0x0Fに設定
 - * 開始アドレスを0x00に設定
 - * データサイズを書き込むデータサイズに設定
 - * 延長時間はデータとコマンドリプライの間に延長を追加して、データをコマンドリプライと区別するのに使う、デフォルト値は3000で、変更できる
- WRITE_FBUFコマンドを発信した後、VC0706のリプライを待ち、リプライが正しければ、VC0706にデータを伝送する。データ伝送完了後、VC0706が再度コマンドリプライを発信して外部マスターMCUにデータ受信完了の通知を出す
- SET_FBUF_LENコマンドを発信してパラメータは画像のサイズで、画像のサイズを設定する
- FBUF_CTRLコマンドを発信し、パラメータは0x03で、書き込んだフレームをカレントフレームに切り替え、表示する

2.5.4. コマンドの詳細説明

詳細説明は” Communication protocol .pdf” をご参照ください。ここでは一部コマンドを例として説明します。

2.5.4.1. GET_FBUF_LEN

コマンドファンクション：FBUFの画像のサイズを取得

コマンドフォーマット：0x56+シリアル番号+0x34+0x01+FBUFフレームタイプ(1バイト)

FBUFフレームタイプ：カレントフレームそれとも次のフレームの画像を取得するかを示す

0:カレントフレームを示す

1:次のフレームを示す

リプライフォーマット：

動作が正しいなら、リプライ内容：0x76+シリアル番号+0x34+0x00+0x04+FBUFサイズ(4バイト)

FBUFフレームタイプのパラメータ設定が間違った場合、リプライ内容：0x76+シリアル番号+0x34+0x03+0x00

例：

0x56+0x00+0x34+0x01+0x00 カレントフレームの画像のサイズを取得

0x56+0x00+0x34+0x01+0x01 次のフレームの画像のサイズを取得

説明：

このコマンドはFBUF画像を読み出す前に使います。FBUFの画像のサイズを取得した後、FBUF画像を読み出しに利用する。

2.5.4.2. SET_FBUF_LEN

コマンド機能：FBUFの次のフレームの画像のサイズを設定する

コマンドフォーマット：0x56+シリアル番号+0x35+0x04+FBUFサイズ(4バイト)

FBUFのサイズは最大65535までである

リプライフォーマット：

0x76+シリアル番号+0x35+0x00+0x00

例：

0x56+0x00+0x35+0x04+0x00+0x00+0xA0+0xB0

FBUFの画像のサイズを0xA0B0に設定する

説明：

設定した画像のサイズはFBUFに書き込んだ画像のサイズと同じでなければならない。異なると、画像の表示にエラーが出る恐れがあります。

2.5.4.3. FBUF_CTRL

コマンド機能：FBUFフレームのコントロール。フレームの停止、切り替または復帰を含む

コマンドフォーマット：0x56+シリアル番号+0x36+0x01+コントロールフラグ(1バイト)

コントロールフラグはFBUFに対するコントロールのタイプを示す

0:カレントフレームを停止

1:次のフレームを停止

2:フレームの更新を復帰

3:フレームを切り替える

リプライフォーマット：

動作が正しいなら、リプライ内容：0x76+シリアル番号+0x36+0x00+0x00

コントロールフラグが間違った場合、リプライ内容：0x76+シリアル番号+0x36+0x03+0x00

例：

0x56+0x00+0x36+0x01+0x00 カレントフレームを停止

0x56+0x00+0x36+0x01+0x01 次のフレームを停止

0x56+0x00+0x36+0x01+0x02 フレームの更新を復帰

0x56+0x00+0x36+0x01+0x0 フレームを切り替える

2.5.4.4. READ_FBIF

コマンド機能：FBIFから画像を読み出す

コマンドフォーマット：0x56+シリアル番号+0x32+0x0C+FBIFフレームタイプ(1バイト)+操作モード(1バイト)+開始アドレス(4バイト)+データサイズ(4バイト)+延長時間(2バイト)

FBIFフレームタイプ：カレントフレームそれとも次のフレームの画像を取得するかを示す

0:カレントフレームを示す

1:次のフレームを示す

操作モード：FBIFデータの伝送方式を示す

Bit0:DMA方式でデータを伝送するかどうかを示す

0:MCU方式でデータを伝送

1:DMA方式でデータを伝送

Bit[2:1]: 11

Bit3: 1

開始アドレス：読み出すデータがFBIFの中の開始アドレス

データサイズ:読み出すデータのバイト数で4の倍数でなければならない

延長時間:コマンドとデータの間の時間間隔で、単位は0.01msである

リプライフォーマット：

動作が正しいなら、リプライ内容：0x76+シリアル番号+0x32+0x00+0x00

その後、データを転送する

転送完了後、リプライコマンドを発信してデータ転送完了の通知を出す。

パラメーターまたは通信が間違った場合、リプライ内容：0x76+シリアル番号+0x32+エラーコード+0x01

例：

0x56+0x00+0x32+0x0C+0x00+ 0x0F+ 0x00+ 0x00+ 0x00+ 0x10+ 0x00+ 0x00+ 0x02+ 0x00+ 0x10+ 0x00

SPIポートを通じてDMA方式でFBIFのカレントフレームの画像を読み出す、画像データの開始アドレスは0x0010、サイズは0x0200、コマンドとデータの間の時間間隔は40.96msである

説明：

システムがコマンドを受信した後、コマンドリプライを行い、FBIFから指定されたサイズのデータを読み出し、SPIで外部マスターMCUに転送します。転送完了後、リプライコマンドを送信してデータ転送完了の通知を出します。

このコマンドを発信する前に指定されたフレームを中止しなければならない。FBIFの画像が更新中にコマンドを発信したら間違った画像を読み出すことになります。

FBIFの画像を読み出す場合、このコマンドで一回全部読み出してもいいし、何回分けて読み出してもいいです。

転送スピードを保証するには、DMAの転送方式がお勧めです。

2.5.4.5. WRITE_FBIF

コマンド機能：FBIFに画像を書き込む

コマンドフォーマット：0x56+シリアル番号+0x33+0x0B++操作モード(1バイト)+開始アドレス(4バイト)+データサイズ(4バイト)+延長時間(2バイト)

操作モード：FBUFデータの伝送方式を示す

Bit0:DMA方式でデータを伝送するかどうかを示す

0:MCU方式でデータを伝送

1:DMA方式でデータを伝送

Bit[2:1]: 11

Bit3: 1

Bit4: FBUFに画像を書き込む操作に今回の操作は一回目の操作であるかどうかを示す

0:一回目ではない

1:一回目

画像を複数回で書き込む場合、一回目で書き込む時、1に設定して、今後の操作で0に設定する

開始アドレス：書き込むデータがFBUFの中の開始アドレス

データサイズ:読み出すデータのバイト数で4の倍数でなければならない

延長時間:コマンドとデータの間の時間間隔で、単位は0.01msである

リプライフォーマット：

動作が正しいなら、リプライ内容：0x76+シリアル番号+0x33+0x00+0x00

その後、データの転送を待つ

データ転送完了後、リプライコマンドを発信してデータ転送完了の通知を出す

パラメーターまたは通信が間違った場合、リプライ内容：0x76+シリアル番号+0x33+エラーコード+0x00
例：

• 0x56+0x00+0x33+0x0B+ 0x0F+ 0x00+ 0x00+ 0x00+ 0x10+ 0x00+ 0x00+ 0x02+ 0x00+ 0x10+ 0x00

SPIポートを通じてDMA方式でFBUFのカレントフレームの画像を書き込む、画像データの開始アドレスは0x0010、サイズは0x0200である

説明：

システムはコマンドを受信した後、関連設定を行います。それから、コマンドリプライを行い、データの受信を待ちます。転送完了後、リプライコマンドを送信してデータ転送完了の通知を出します。

このコマンドを発信する前に指定されたフレームを中止しなければならない。

転送スピードを保証するには、DMAの転送方式がお勧めです。

以上。