

農業用ドローンにおける画像処理を想定したインターフェース開発とその評価 Image processing interface development and evaluation in agricultural drones

中川 諒¹⁾ 山口 佳樹²⁾
Ryo Nakagawa Yoshiki Yamaguchi

1 はじめに

世界人口は 2020 年に 78 億に迫り、2050 年には 100 億を超える可能性が示唆されている [1]。これをまかなう世界の耕地面積は 2009 年時点で約 1,527 万 km² であり、内訳をみると、灌漑用地は 1961 年比で 17% 程度の増加を示しているが天水用地はほぼ横ばい (1961 年比で -0.2%) で推移している [2]。灌漑用地と天水用地の比は 1:4 であり、世界人口の増加に合わせた耕地面積の拡大は難しいことがわかる。

さらに、一人当たりの収穫面積 (耕作面積のうち収穫または出荷した面積) を見ると、1961 年は 21.1 km² であったが、2019 年には 9.5 km²、2050 年には 7 km² 程度にまで減少すると予想されている [3]。国連によって推進されている持続可能な開発目標 (SDGs: Sustainable Development Goals) では、将来の深刻な食糧不足を避けるため、本問題を喫緊を要する課題の一つとして設定している。この解決に向け、現存する耕地の利用効率および単位面積あたりの収穫率を高めることも議論されており、情報技術を導入したスマート農業の利活用が注目を集めている。

そこで本稿では、我が国のスマート農業における、ドローンの応用利用について検討する。これまでに、DJI 社のドローンの利用 [4][5] やマルチスペクトルカメラの利用 [6] などが報告されているが、いずれも特定の品種に対しての応用であり、ドローンに特定のセンサを組み込んだ提案となっている。これは、欧米などの大規模農業形態には適しているが、様々な品種を狭い耕地に作付けする日本の農業とは合致しないことが多い。つまり、我が国のスマート農業におけるドローン利用を考えると、空撮において要求される以下の能力

- ・演算はリアルタイムで処理を終えること
- ・リアルタイムで演算結果を地上で確認できること
- ・演算に要する消費電力が少ないこと
- ・空撮後、改めて全データを確認できること

に加えて、

- ・空撮中、撮影対象の作物に合わせて、演算を即座かつ自由に変更できること

が加わる。このため、空撮中の確認内容に応じて演算の変更が可能であり、変更に関らず全演算を実時間内に完了できる演算装置が望まれる。しかし、全演算に対して十分な性能を持つデバイスを採用すると、消費電力が非常に大きくなり、要求性能を満たすのは難しい。そこで本稿では、演算装置に FPGA を採用した図 1 に示すドローンシステムを提案する。

- 1) 筑波大学大学院システム情報工学研究群, Graduate School of Science and Technology, University of Tsukuba
- 2) 筑波大学システム情報系, Faculty of Engineering, Information and Systems, University of Tsukuba

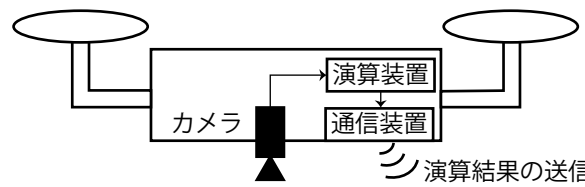


図 1 ドローンシステムの概要図。本発表範囲はカメラと演算装置とのインターフェース部分。

ここで、高齢化が進む農業分野において、新機材および新システムの導入は難しいという声も大きい。このため本稿では、ドローンに市販のデジタルスチルカメラを搭載し、空撮することを仮定した。

市販のデジタルスチルカメラと FPGA の組み合わせでリアルタイムに撮影画像を確認するためには、カメラから FPGA へ遅滞なく送る機構が必要となる。そこで、多くのデジタルスチルカメラが持つ SD (Secure Digital) メモリカードに着目し、FPGA 上で SD カードをハードウェアエミュレートする演算回路を準備し、ここで問題となっている実装上の課題を解決することとした。

具体的には、本稿で使用するデジタルスチルカメラ (Canon PowerShot G9X Mark.ii) の SD カードスロットに延長ケーブルを接続し、その反対側の端子と FPGA を接続する。デジタルスチルカメラは、FPGA 上のエミュレーション回路を SD メモリとしてみなし、データを書き込む。FPGA は、書き込まれたデータを別に接続した SD カードに保存するとともに、直接演算処理してユーザーにリアルタイムで出力する。

本論文の構成は以下の通りである。第 2 章で SD カード規格について述べ、第 3 章で本稿で提案するシステムを、第 4 章でその結果について示す。続く第 5 章で結果に対しての考察を行い、第 6 章で本稿をとじる。

2 SD カード規格

SD カードには、標準的な SD (SDSC) のほかに SDHC や SDXC, UHS-I や UHS-II などの規格があり、最大容量やコマンド体系などが異なっている。本稿では取り扱うハードウェアの都合上、主に SDSC カードと SDHC カードの規格について解説する。

2.1 電氣的仕様

SD カード [7] は、SD モードという 1 本のクロックと 1 本のコマンド信号、4 本のデータ信号による方法でホストとカードの間の通信を行う。

図 2 に SD カードの各端子の配置と機能を示す。コマンド端子ではホストからカードに対する初期化やデータの読み出し・書き込みなどの操作とカードからのレスポンスの通信を行い、データ端子ではカードから読み込むデータおよび書き込むデータの通信を行う。

コマンドとデータの送受信はいずれもシリアル通信で

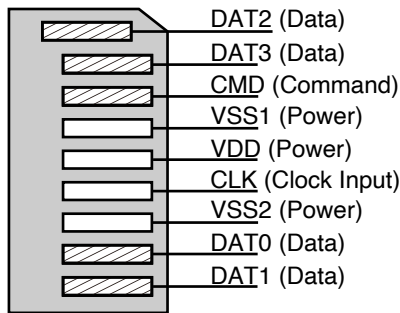


図2 SDカードの端子配置(端子面)と各端子の機能。コマンドとデータ端子(斜線部)は双方向で使用される。クロックおよび電源はホスト側から供給。

あり、ホストが送るクロックの立ち上がりエッジに同期して行われる。クロック周波数は可変であり、初期化時には400kHzで、データ送受信時は本稿の範囲では50MHzになる。さらに通信を行わないときはホストはクロックを停止し、カードをスリープ状態にすることができる。

コマンド線およびデータ線は送信・受信ともに同一の線で通信する双方向バスであり、例えばホストがコマンドを送信中の場合はカードがコマンド線をハイインピーダンスにセットして、ホストからのコマンドを受信する。ホストが定められた手順でカードの初期化を行うことによって、SPI (Serial Peripheral Interface) モードによる通信を行うことも可能である。この場合は双方向バスを使用せずに2本の単方向バスでのシリアル通信が可能となるため送受信の切替えが不要になり実装が容易になるが、本稿では接続するカメラがSPIモードに対応していないため取り扱わない。

2.2 コマンド・レスポンス・データ

図3にコマンド・レスポンス・データのフォーマットを示す。SDバスのコマンド(CMD)は48ビットの固定長である。コマンドの開始はクロックに同期してコマンド線を0にすることによって識別され、コマンドの最後は必ず1である。コマンドにはインデックス、引数、CRC7(巡回冗長検査)チェックのフィールドが含まれる。CMD55に続くコマンドは、アプリケーションコマンド(ACMD)として解釈される。例えばCMD55の後にCMD41が送信された場合は、ACMD41となる。一部のコマンドを除いて、ホストからコマンドを発行するとカードからレスポンスが得られる。レスポンスは48ビットおよび136ビットの2種類があり、コマンドによって異なる。

データ通信はSingle Block Read, Multiple Block Read, Single Block Write, Multiple Block Writeの4種類がある。読み出し・書き込みのブロック長はCMD16によって変更可能である。本稿で製作したシステムでは512バイトとなっている。Multiple Blockに対する命令は、CMD12が発行されるまで続けられる。各データチャンクにはCRC16のフィールドがある。読み出しと書き込みのアドレス発行はSDSCではバイト単位、SDHCではブロック単位で行われる。

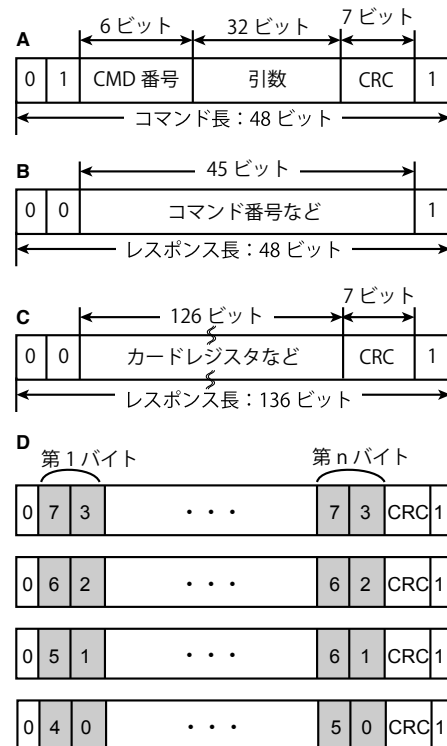


図3 A: SDのコマンド書式。B, C: SDのレスポンス書式。D: SDのデータ書式。各データ書式の先頭の0と末尾の1はそれぞれstart bitとstop bit。(D)は4本のデータ信号でグレーの2列が1バイトを構成している。

2.3 カード初期化シーケンスの調査

ホストは電源投入後に、所定の手順でコマンドを発行し、カードを初期化する。この際にホストはレジスタを読み出し、カードの情報を認識する。しかし使用するカメラが実際にどのようにコマンドを発行して初期化をするのか不明であったため、SDカードとカメラの間の信号をロジックアナライザに接続してコマンドとレスポンスの内容を調査した。

図4に示すようなSDカードアダプタを介してカメラに2GBのSDカードを挿入し、アダプタ上の端子をロジックアナライザ(Zeroplus社LogicCube Pro)に接続してカード初期化時のSDバスのコマンドとデータをプロトコルアナライザで解析した。

またCMD6に対するカードのレスポンスから、実際

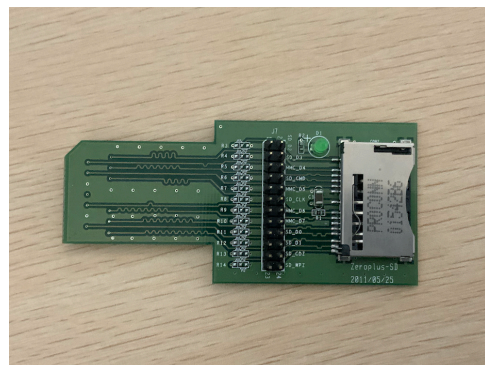


図4 SDカードアダプタ。左側をカメラのSDカードスロットに、右側をSDカードに挿入。

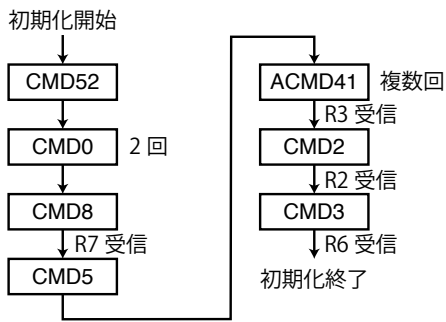


図5 カメラ初期化時にカードに送信されるコマンドとそのフローチャート。"CMDn", "ACMDn"はカメラからカードに送信されるコマンドとその番号(n), "Rn"はカードからカメラに戻る応答の種類を表している。

のSDカードに設定されているカードの性能に関するレジスタの値を得ることができた。

図5に解析の結果得られた初期化のフローチャートを示す。

2.4 データ読み書きの調査

カメラはSDカードの初期化が終了すると、カードのレジスタにアクセスして容量などの情報を取得する。この情報をもとにファイルシステム領域のデータを読み書きする。ファイルシステムの操作に関しては規格に定められていないため、実際のカメラの動作を解析する必要があった。そのため、2.3節と同様の構成で、初期化直後の読み出し時にアクセスする領域のアドレスと内容を解析した。さらに撮影後に画像をカードに書き込む動作も解析し、カメラがカードに対して行う一連の操作を把握した。図6にアドレスマップを示す。

まず初期化直後のファイルシステムを調べる動作を解析した。その結果、以下のような手順でファイルシステムを調べていることが明らかになった。

1. カードのメモリ領域の先頭部分に記録されている、BIOS Parameter Block (BPB) 領域にアクセスし512バイトを読み出す
2. BPBに記録されているFAT領域の先頭アドレスの値をもとに、FAT領域にアクセスし読み出す
3. ディレクトリ領域を読み出し、すでに記録されているディレクトリやファイル調べる

次に撮影後に画像を書き込む動作を解析した。その結果、以下のような手順で画像をカードに書き込んでいることが明らかになった。

1. FATのディレクトリエントリを読み出し、保存するファイル名を特定する
2. ディレクトリ"101_01"の開始クラスタを調べる
3. ディレクトリ領域の情報に基づいて、適切なデータ領域のクラスタ内にデータを書き込む
4. 2つのFAT領域に対して書き込みを行い、画像を書き込んだクラスタに使用中のラベルを付ける
5. 新たに作成した画像ファイルのディレクトリエントリにファイル名などの情報を書き込み、ファイルの作成を完了する

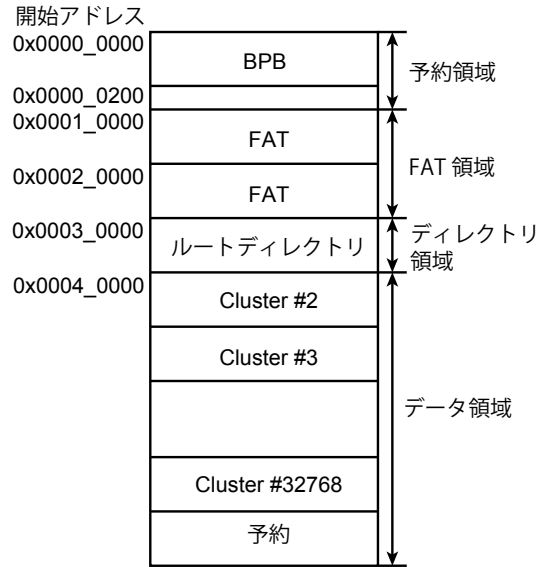


図6 実装したFAT16ファイルシステムのアドレスマップ (文献[8][9]をもとに作成)

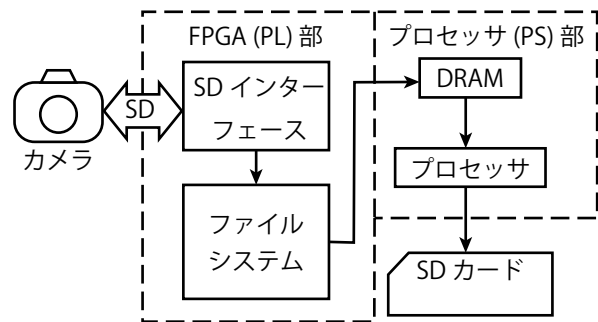


図7 提案するシステムの構成図

3 提案するシステム

本章では、インターフェースシステムを構成するために必要なSDインターフェース回路・ファイルシステム回路・ソフトウェアの3つについて説明する。

3.1 FPGAの使用

本稿では画像入力インターフェースを構成するためにFPGAを使用する。その理由は、SDカードはカメラから入力される最高50MHzという比較的高速なクロックに同期して、規定のタイミングで状態遷移や応答を返すなどの処理をする必要があるためである。また、取得後の画像処理ではドローン上での使用が想定されていることから、性能および消費電力の観点でFPGAを使用する予定である。したがって画像取得インタフェースという観点からもFPGAの利用が適している。

FPGAにはXilinxのZynq-7020 SoC (XC7Z020-1CLG400C)[10]を搭載したZYBO-Z20ボード(Digilent)を使用する。このFPGAは同一パッケージ内に再構成可能な論理回路部分(PL)と、デュアルコアのARMプロセッサ部分(PS)を搭載している。使用するボードではSoCに512MBのDDR3 SDRAMを搭載しており、PLからPSを介してDRAMにアクセスすることができる。

SDカードのホスト機能のIPは商用・オープンソースを含めて多種あり容易に実装が可能であるが、今回はカ

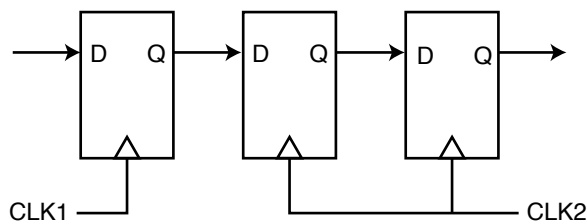


図8 CDC概念図。多段フリップフロップで信号を受け取り、異なるクロックドメイン間で信号の受け渡す。

メラをホストとしてSDカードのカード側の機能を実現する必要があり、この機能を実現したIPは1つのみ事例がある[11]。さらに画像の入力という必要な機能のみ実装するため、当該先行事例のIPを用いない方法で今回のインターフェースを作成した。

3.2 SD インターフェース回路

SDカードのインターフェースは、ホストから供給されるクロックによって駆動される。しかしSDカードホストから供給されるクロックは周波数がkHzからMHzで可変であり停止することもあるため、FPGA内部の論理回路の駆動には適していない。そのため、FPGAの内部回路はFPGAボード上で生成されたクロックによって駆動した。したがって、2つのクロックの間でデータの受け渡しをするClock Domain Crossing (CDC)が必要となる。CDCではフリップフロップのセットアップ・ホールド時間を満たさないタイミングで信号が変化する可能性があり、これは不安定なメタステーブルの発生につながる。したがってCDCの場合は2-3段程度のシフトレジスタを構成し、メタステーブルが発生しても伝搬しないように実装する手法が取られる。

SDカードのインターフェースでは双方向バスが用いられている。この場合、トライステートバッファによって送信あるいは受信を切り替えなければならない。本稿で製作した画像入力インターフェース回路では画像の入力だけではなく、カメラがSDカードに対して行う初期化の処理や、カメラからの書き込みなどのコマンドに対してSDカードの規格に従ってレスポンスやデータを返す必要があるため、出力の機能も必要である。コマンドによってレスポンスやデータ応答の要不要や長さが異なるため、入出力の切り替えはコマンドの内容を見て定められたタイミングで行わなければならない。

3.3 ファイルシステム回路

SDカードではFATファイルシステムの使用が一般的である。そのため、本稿で開発するシステムでもFATファイルシステムを使用する。提案手法ではファイルシステムの領域全体をDRAMに実装することも可能であるが、その場合多くのメモリ領域が必要となり、他の処理で使用するメモリ領域が不足してしまう。したがってDRAMは画像を書き込むために最低限必要な量のみを確保して、その他のファイルシステムに関する情報が記録されている予約領域・FAT領域・ディレクトリ領域(図6参照)は論理回路としてハードコーディングし、ROM化する方針をとった。

ハードコーディングしたファイルシステムはFAT16

である。FAT16を採用した理由は、FAT32ファイルシステムと比較して構造が単純であるため、ハードコーディングが容易であると判断したからである。

書き込みトランザクションから画像ファイルを検出し抽出するためには、カメラがカードに書き込んでいるデータを画像ファイル・ディレクトリエントリ・FAT領域の3種類から区別する必要がある。2.4節の書き込み動作解析から、書き込み中にアドレスがデータ領域に対して発行され、かつJPEGまたはRAW画像ファイルの先頭にあるマジックナンバーを検出した場合に画像ファイルの書き込みと判断し画像書き込み状態に移行することとした。マジックナンバーは、JPEGの場合は"FF D8 FF E1"で、RAW画像の場合は"49 49 2A 00"である[12]。

3.4 ソフトウェア

DRAMに書き込まれた画像データを外部メディアに記録するために、Zynq-7020 SoCのARMプロセッサで動作するソフトウェアを使用した。ファイルシステム回路でDRAMへの画像の書き込みが完了したことを検出できるため、その信号を割り込みのトリガーとしてDRAM上の画像ファイルを読み出し、SDカードに保存するようにした。

4 システムの実装

本章では、3章で述べたシステムについて、その実装方法について述べる。

4.1 ハードウェア実装

第3章で述べたシステムのハードウェアをVerilog HDLで実装した。

第3章で述べたようなSDカードの機能をすべて実装することは容易ではない。したがって本稿では、スマートフォン用に暗号化プロセッサなどを搭載したセキュアなSDカードを開発する"Vault"というGoogleから派生したプロジェクト[13]が開発したオープンソースのSDカードエミュレーションモジュールを使用した。以降このモジュールを「SDHCモジュール」と呼称する。

SDHCモジュールはClass 10のSDHCカードとして振る舞う。外部インターフェースとしてSDカードのインターフェース、内部データバスとしてWishboneバス[14]をもち、ホストからカードへのデータの読み書きはWishboneバスに接続されたメモリに対して行われる。

SDHCモジュールの大まかな構造は図9のようにになっている。図中灰色で示した部分はカメラから入力されるSDのクロックで動作し、白色で示した部分はFPGA内部で生成する50MHzのクロックで動作する。sd_mgr内に斜線で示す非同期RAMは2種類のクロックそれぞれで読み書きが可能である。

SDHCモジュールは、これまでのカメラの動作分析をもとにテストベンチを作成し、その動作を調べた。その結果、Multiple Block Readコマンド発行時にWishboneバスのアドレスが4ずつ増加していることを確かめた。Wishboneバスのデータ信号の幅が32ビット、すなわち4バイトであることから、アドレスはバイト単位で発行されているとわかった。

FPGAの論理回路からZynq SoCのDARMにアクセスするためには、AXIバスを使う必要がある。そのため、

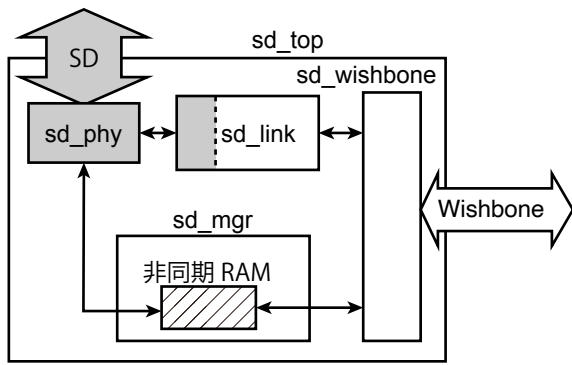


図9 SDHC モジュールの概略図

SDHC モジュールの Wishbone バスから AXI バスに変換するモジュールをファイルシステム回路の後段に追加した。

画像ファイル以外のデータを取り込まないようにするために、待機状態を設けた。これはファイルシステムの予約領域・FAT 領域・ディレクトリ領域に対してアクセスするための状態で、これらの領域に書き込みをしてもデータは破棄されるようにした。画像書き込み状態から待機状態への遷移は、これらの領域に対してアドレスが発行された場合になされるようにした。

SDHC モジュールをそのまま使用して実装したシステムでは、カメラによって異常なカードであると判断された。原因を探るためにロジックアナライザで FPGA・カメラ間の通信を解析すると、クロックが 50MHz に変更されてからコマンド及びレスポンスの CRC エラーが多発し、また解析で得たデータパケットにも CRC エラーが起きていることがわかった。またクロックの波形を観測すると、ジッタが見られた。

以上の解析から、SDHC モジュールに設定されているデータ転送速度が速いことが信号品質劣化を招いていると考え、SDHC モジュールのカード情報に関するレジスタを書き換えることで、遅い種別のカードであるとカメラに判断させることにした。SD カードのデータ転送速度に関するパラメータは、カードの CSD レジスタに記録されている。そこで、スピードクラスに関する値やデータアクセス時間に関する TAAC などの値を書き換えた。書き換える値は、2.3 節のロジックアナライザでの解析で取得した 2GB の SDSC カードのものを利用した。

ファイルシステム回路は SDHC モジュール側データのバイトオーダーがビッグエンディアンであり、Zynq-7020 SoC の DRAM がリトルエンディアンであることからバイトオーダーを並び替える機構を挿入した。FAT16 ファイルシステムは、事前評価のときと同様にイメージファイルを作成し、マウントしてカメラに必要なディレクトリを作成した。作成したイメージファイルは Verilog HDL の case 文として実装できるように加工し、ファイルシステム回路中に埋め込んだ。

ハードウェアを論理合成・配置配線した後の FPGA のリソース使用量を表 1 に示す。SDHC モジュール内部でデュアルポート RAM が使用されているため、BRAM の使用量がおおよそ 25% と大きくなった。

表 1 FPGA のリソース使用量および使用率
BRAM の消費が多くなっている。

リソース	使用量	使用可能量	使用率 (%)
LUT	6019	53200	11.31
LUTRAM	705	17400	4.05
FF	9204	106400	8.65
BRAM	32	140	22.86
IO	11	125	8.80
BUFG	4	32	12.50

4.2 ソフトウェア実装

Zynq の PL に実装したハードウェアは DRAM への書き込みのみ行うため、書き込んだデータを取り出すために CPU を使用した。CPU の処理は主に表 2 に示す 4 つに分けられる。

プロセッサで動作するソフトウェアを C 言語で実装した。プロセッサは各ペリフェラルを初期化するとファイルシステム回路からの画像書き込み完了割り込みを待機するようし、割り込みが発生すると DRAM から画像ファイルを読み込んで SD カードに転送するようにした。

ファイルシステム回路は DRAM に書き込んだワード数をカウントしており、これに 4 を乗じることで書き込んだバイト数を得ることができる。しかし DRAM に書き込まれるデータは、FAT ファイルシステムのクラスタ境界にアライメントされているため、書き込んだバイト数と実際の画像ファイルのサイズが一致するとは限らない。そのため、割り込みでファイルシステム回路から通知された書き込みワード数から DRAM に書き込んだデータの末尾を求め、そこから図 10 のようにメモリアドレスが減少する方向に DRAM を 1 バイトずつ走査して JPEG および RAW 画像の末尾を示す“FF D9”のパターンを見つける処理を実装した。これによりファイル末尾を検出し、1 バイト単位で正確にファイルサイズを求めるようにした。

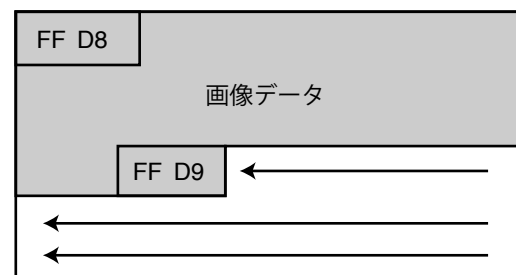


図 10 画像の末尾を検出するための走査のイメージ図

5 評価と検証

画像保存用の 2GB の SD カードを FPGA ボードの SD カードスロットに挿入し、カメラは図 11 に示す SD 延長ケーブルを介して FPGA ボードと接続した上でシステムを動作させた。デジタルカメラの表示で、撮影可能枚数が表示されカードが異常であると判定されず初期化が完了したことを確認した。また、設定メニューから 1.9GB の SD カードとして認識されていることを確認した。

表2 ソフトウェアの機能

処理	内容
各デバイスの初期化	HWFS の設定レジスタと割り込みコントローラ, SD カードインタフェースを初期化する
割り込みの処理	画像書き込み終了の通知を HWFS モジュールから受け取る
画像サイズの計算	HWFS モジュールから通知された書き込みワード数を基にメモリを走査して画像ファイルの末尾を検出しサイズを計算する
SD カードへのファイル書き込み	画像ファイルを DRAM から SD カードに複製する



図11 SD エクステンションケーブルの画像。両端がSDカードの形状をしているフラットケーブルである。

初期化完了後に JPEG 撮影モードに設定したデジタルカメラで撮影をすると, FPGA ボード上の LED が点灯し DRAM に書き込みが行われていることを確認した。転送終了後もカメラは製作したシステムを正常なカードであると認識していた。

画像ファイルの書き込み終了後に SD カードを抜き取り, PC の画像ビューアで書き込まれた画像ファイルを表示して目視で正常に画像が記録されていることを確認した。

さらに, 画像ファイルをバイナリエディタで開いて JPEG 画像ファイル特有の先頭と末尾のマーカを確認し, 先頭には "FF D8" があり末尾には "FF D9" が記録されていることを確認した。図に先頭部分をバイナリエディタで表示した様子, 図に末尾部分をバイナリエディタで表示した様子を示す。

以上から, 製作したインターフェースシステムがデジタルカメラに対して SD カードとして動作していることを確認できた。また, カメラからインターフェースシステムに書き込まれる画像データから正確に画像ファイル

を抽出し, それを外部に接続した SD カードに保存できていることも確認できた。

```

+0 +1 +2 +3 +4 +5 +6 +7 +8      : 7A 8E BF 9D 05-09 2C A0
0 FF D8 FF E1 6F FE 45 78-69    ! E7 F0 A7 B1 3C-C3 36 0D
10 08 00 00 00 0D 0E 01-02     : 9F 4A 8C 27 CA-70 58 6D
20 00 00 0F 01 02 00 06 00-00   ? 93 4A F7 34 47-FF 09

```

図12 画像をバイナリエディタで確認した結果。左：先頭部分に FF D8 が確認できる。右：末尾部分に FF D9 が確認できる。

6 まとめ

組み込み機器用の画像出力インターフェースを持たない一般的なデジタルカメラから, SD カードのインターフェースを用いることによって撮影した画像を取得可能であることを示した。本稿で製作したシステムをドローンに搭載してリアルタイム画像処理をするためには, 動画撮影中にカードに保存されるデータを読み出しながら画像処理回路に送る必要がある。また, 本稿の SDHC モジュールに関する成果は新しいアーキテクチャの FPGA に搭載された。Zynq-7020 SoC 以外での利用も想定し, 今後は RAM の使用量を減らしつつリアルタイムで映像データを取り出す機構について検討していきたい。

参考文献

- [1] 国連人口基金. 世界人口白書 2020, 2020.
- [2] The Food and Agriculture Organization of the United Nations (FAO). *The State of the World's Land and Water Resources for Food and Agriculture*, ROME edition, 2011. (Summary Report. Figure 1.).
- [3] 農林水産省. 食料・農業・農村白書, 平成 21 年度, 2010 年 6 月. (世界の穀物の生産量, 収穫面積, 単収等の推移と見通し (図 1-10) より) .
- [4] DJI Stories - 「世界を変える農業」. <https://www.youtube.com/watch?v=RNdXkM3PmW8>, 2016.
- [5] P4 Multispectral - DJI. <https://www.dji.com/jp/p4-multispectral>.
- [6] スペクトル技術による次世代農業の可能性. <https://smartnogyo.com/archives/221.html>, 2018.
- [7] SD Card Association. SD Specifications Part 1 Physical Layer Specification Simplified Specification Version 7.10. 2020.
- [8] FAT ファイルシステムのしくみと操作法. <http://elm-chan.org/docs/fat.html>, 2020.
- [9] Microsoft Corporation. FAT: General Overview of On-Disk Format. No. Generic, 2000.
- [10] Xilinx Inc. Zynq-7000 SoC テクニカルリファレンスマニュアル. 2015.
- [11] SD Memory Slave Controller. <https://japan.xilinx.com/products/intellectual-property/1-lulofg.html>.
- [12] JPEG ファイルの構造. <https://hp.vector.co.jp/authors/VA032610/JPEGFormat/StructureOfJPEG.htm>.
- [13] ProjectVault. ProjectVault/orp. <https://github.com/ProjectVault/orp>, 2015.
- [14] OpenCores Organization. WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores. No. Generic, 2002.