

## 1. プログラムの作製

以前のミーティングにて話したようにオプティカルフローセンサ及びUWBの全てのセンサ値を取得できるようになったため、それらを組み合わせた位置推定用のプログラムを作製中である。カルマンフィルタを用いた位置推定のプログラムの一部は以前に赤堀さんが作製したものをそのまま流用し、加筆、修正する形で用いてきた。しかし、赤堀さんは処理を関数やクラスにまとめることはほとんどなく、上から処理を順番に書いていくタイプであったので、変数の受け渡しや、関数間の繋がりが非常に分かりにくかった。したがって、今回大きなプログラムを書き直すにあたり、それぞれの処理を別の関数にわけ、グローバル変数をあまり多用しないものに改めながら、書き進めている。また、一つ一つの関数にコメントを付け加え、それぞれの関数がどういった処理を行っているのかを見分けるような工夫も行っている。現在の進捗度は2種類のセンサ値を組み合わせた位置推定プログラムは完成した。しかし、事象でも述べる通信エラーの問題のため、意図した動作は行えていない。

## 2. I2C通信における不具合

以前にも述べた通り、I2C通信においてUWBの値をRaspberry Piに送った際に上位Byteと下位Byteが混ざって値がおかしくなるといった症状は改善できた。しかし、I2C通信中にRaspberry Piが「IOError: [Errno 5] Input/output error」という通信エラーを吐いてしまい、通信がストップすることが頻繁にある。また、それらの発生するタイミングは開始してから数秒から数分と様々である。現在考えられる原因はハード側、ソフト側の2通りがある。1つ目はノイズが影響するハードの問題である。調べたところ、I2C通信は基板上でのIC間の通信に用いるように開発された規格であり、長距離の通信にはあまり向いておらず、長距離になるとコードにノイズが乗り、信号波形がなまり（角が丸くなる）、信号をうまくうけとれなくなるということがわかった。防ぐためにはコードを短くすることや、コードをアルミホイルなどで包んでシールドすることが有効らしい。さらに2つ目の原因是前回にも述べた通り、Raspberry Pi側にて設定可能なI2Cのポーレートのや通信する際の遅延（delay）の設定が最適に調整できていおらず、通信の際に発生するタイミング的な問題であると考えている。一度オシロスコープを用いて波形を確認し、波形がなまっているか、またエラーが発生した際にどのような波形になっているかを確認しなければならないと考えている。

## 3. フライトコントローラ（Pixracer）を用いたPWM信号の制御

飛行ロボコンにて救援物資を投下するといったミッションがあるが、今までPixracerにて送信機からの指令値を受け取り、その信号をRaspberry Pi zeroに送信し、値に応じてPWM信号を生成してサーボモータに送り、角度制御するというものであった。したがって、2つのコンピュータ間の通信が遅いことやRaspberry Pi zeroの処理能力が低いことが原因で送信機のスイッチを操作してから物資が投下されるまでに1秒ほどの遅延があり、箱に投下するのが多少難しいとされていた。そこで、Pixracerからブラシレスモータを制御する以外のPWMを直接出せないかと調べたところ、その方法が存在した。本来はUAVを用いて救援物資や配送の荷物を目的地にて投下することや、カメラのジンバルを制御するのを目的に開発されたものである。Pixracerには元々6つのPWM信号が出力可能なピンがある（Fig.1）。

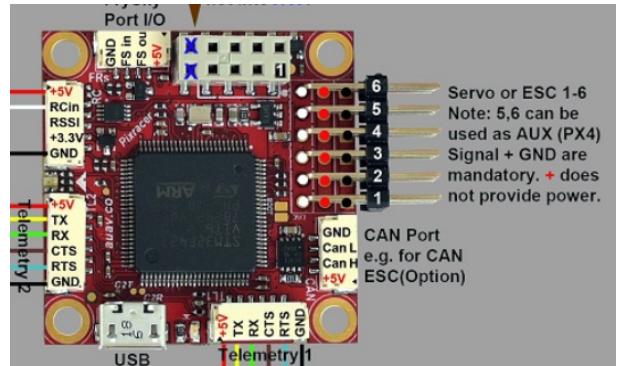


Fig.1: Pixracer のピン配置 [1]

図から分かるように1から4まではクアッドなどのブラシレスモータを制御するためのピンであるが、5,6ピンに関してはAUXピンであるため、ブラシレスモータ以外の飛行に関係のない機器をPWM信号を用いて制御可能である。なお、1から4ピンに関してはESCを制御するため、400Hzで更新されるが、5,6ピンは一般的なサーボモータと同様の50Hzで更新されるため、サーボモータを制御するのに適している。また、Pixracerからサーボモータを動かすほどの電流を取ることは不可能なため、1から6ピンまでのいずれかの中央の+ピンに電源を流せば、内部で繋がっているため、サーボモータに電源を供給することが可能である（Fig.2）。

また、Mission Planner（Pixracerのパラメタを書き換えるソフト）を用いてリリース時とピック時のPWM幅も変えられるため、自由にサーボモータの角度も調整することが可能である。この機能を用い、サーボモータを制御したところ、今までスイッチ操作から動き出



Fig.2: サーボモータと Pixhawk の接続図 [2]

すまでに 1 秒ほどかかっていたものがほぼ遅延なしにサーボモータの角度を制御可能になった。将来、これらをうまく活用すれば、別のマイコンにて PWM を読んで何かを制御するなどといった事が可能ではないかと考えている。さらに山崎が以前紹介した市販の EPM (Fig.3) は PWM 信号や CAN 通信を用いて制御可能であり、ArduPilot のフォーラムでもあらかじめ GPS で設定した地点に救急キットを落とす例を紹介していた (Fig.4)[3]。

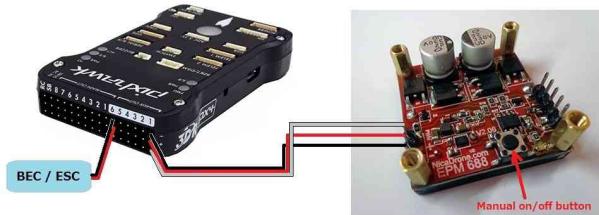


Fig.3: EPM と Pixhawk の接続図 [3]



Fig.5: 救急キットとクアッドロータ [3]

## 参考文献

- [1] Pixracer, [https://docs.px4.io/v1.9.0/en/flight\\_controller/pixracer.html](https://docs.px4.io/v1.9.0/en/flight_controller/pixracer.html)
- [2] Servo Gripper, <http://ardupilot.org/copter/docs/common-gripper-servo.html>
- [3] Electro Permanent Magnet Gripper (EPM688), <http://ardupilot.org/copter/docs/common-electro-permanent-magnet-gripper.html>
- [4] Nica Drone, <https://nicadrone.com/products/epm-v3>



Fig.4: 救急キットの固定方法 [3]

これらを参考に将来、天井に対して EPM を用いて機体を吸着させる際にも専用の受信機をクアッドに搭載することなく吸着と離脱などを制御可能であると考えている。ちなみにこの EPM は「OpenGrab EPM V3 R5C」という名前で 159.99 ドルにて販売されている [4]。

## 4. 今後の予定

- ・ プログラムの通信エラーの解消
- ・ 位置推定値の妥当性の検証