

FreeMCOM for Airsoft v1.0.0 導入説明書

FreeMCOM for Airsoft development squatron

2014 年 11 月 24 日

目次

| | | |
|---------|---------------------------|----|
| 第 I 部 | これは何ですか？ | 3 |
| 第 II 部 | セットアップ | 4 |
| 1 | セットアップ | 4 |
| 1.1 | 基本的な使用に必要なハードウェア | 4 |
| 1.2 | 基本的な使用に必要なソフトウェア | 4 |
| 2 | あなたの M-COM を組み立てる | 5 |
| 2.1 | 規定値の調整について | 5 |
| 2.2 | 組立について | 5 |
| 第 III 部 | ゲームでの使用方法 | 8 |
| 3 | ゲームで使う | 8 |
| 3.1 | 遠隔監視を使用しない場合の使用方法 | 8 |
| 3.2 | 遠隔監視を行う場合の使用方法 | 8 |
| 第 IV 部 | トラブルシューティング | 9 |
| 4 | うまく動かない時は | 9 |
| 4.1 | 遠隔監視を使用しない場合の確認事項 | 9 |
| 4.2 | 遠隔監視を使用する場合の追加の確認事項 | 9 |
| 第 V 部 | 本プログラム及び使用ライブラリのライセンス、著作権 | 10 |
| 5 | 本プログラムの著作権及び使用ライセンス | 10 |
| 5.1 | 使用ライブラリの著作権及びライセンス | 10 |

第 I 部

これは何ですか？

サバゲ・ペイントボール等で、Battlefield(R) シリーズの「ラッシュ」に準じたルールで遊ぶための、Arduino を使用した M-COM のクローンを製作するためのソフトウェアです。要するに、

- 予め Arduino に本ソフトウェアを書き込んだ上で、ボタン・ブザー・赤色灯を接続しておく
- 攻撃チームと防衛チームに分かれる
- 攻撃チームは M-COM を破壊するために、ボタンを長押しして起爆の準備をし、爆発まで防衛チームに解除されないように守らなくてはなりません。
- 防衛チームは攻撃チームを排除すると共に、万が一起爆準備をされてしまった場合、ボタンを長押ししてそれを解除しなくてはなりません。

第 II 部

セットアップ

1 セットアップ

1.1 基本的な使用に必要なハードウェア

1. Arduino (リレーシールドとの兼ね合い上、Arduino Uno/Leonard 等、標準サイズ推奨。)
2. 大音量のブザー
3. LED 赤色灯
4. 押ボタンスイッチ (モーメンタリ) ×1 (起動/解除用)
5. タクトスイッチ ×1 (リセットボタン用)
6. 10k Ω 抵抗 (プルダウン抵抗) ×1
7. ユニバーサル基板 ×1
8. リレーシールド (本書では SLD01101P を想定)
9. 電源 (Arduino 用と、LED 赤色灯・ブザー用の 2 系統が必要)
10. ケース (ダイソーの適当な箱を電動ドリルで加工すれば事足ります)

1.1.1 遠隔監視を行う場合に追加で必要となるハードウェア

1. PC
2. XBee ×2 (モデルにより外部アンテナも必要)
3. XBee エクスプローラ ×1
4. ワイヤレスプロトシールド ×1 (本解説書では ARDUINO-A000064 を想定)

1.2 基本的な使用に必要なソフトウェア

1. Arduino IDE 1.0.5 以降 (<http://www.arduino.cc/>)

1.2.1 遠隔監視を行う場合に追加で必要となるソフトウェア

1. Python 2 系列 (2.7 以降)
2. wxPython(導入している python のバージョンに対応するもの)
3. pyserial(導入している python のバージョンに対応するもの)
4. X-CTU

なお、本文書では関連ソフトウェアのインストールについては取り扱いません。必要に応じインターネット上に多数存在する解説サイトをご覧ください。

2 あなたの M-COM を組み立てる

2.1 規定値の調整について

1. 好みに合わせ、freemcom/freemcom.ino の 26 行～29 行辺りのグローバルな変数・定数を変更します。
これは、`struct TIMER { const long LIMIT , const long STEP , long TIME }` という構造体になっています。

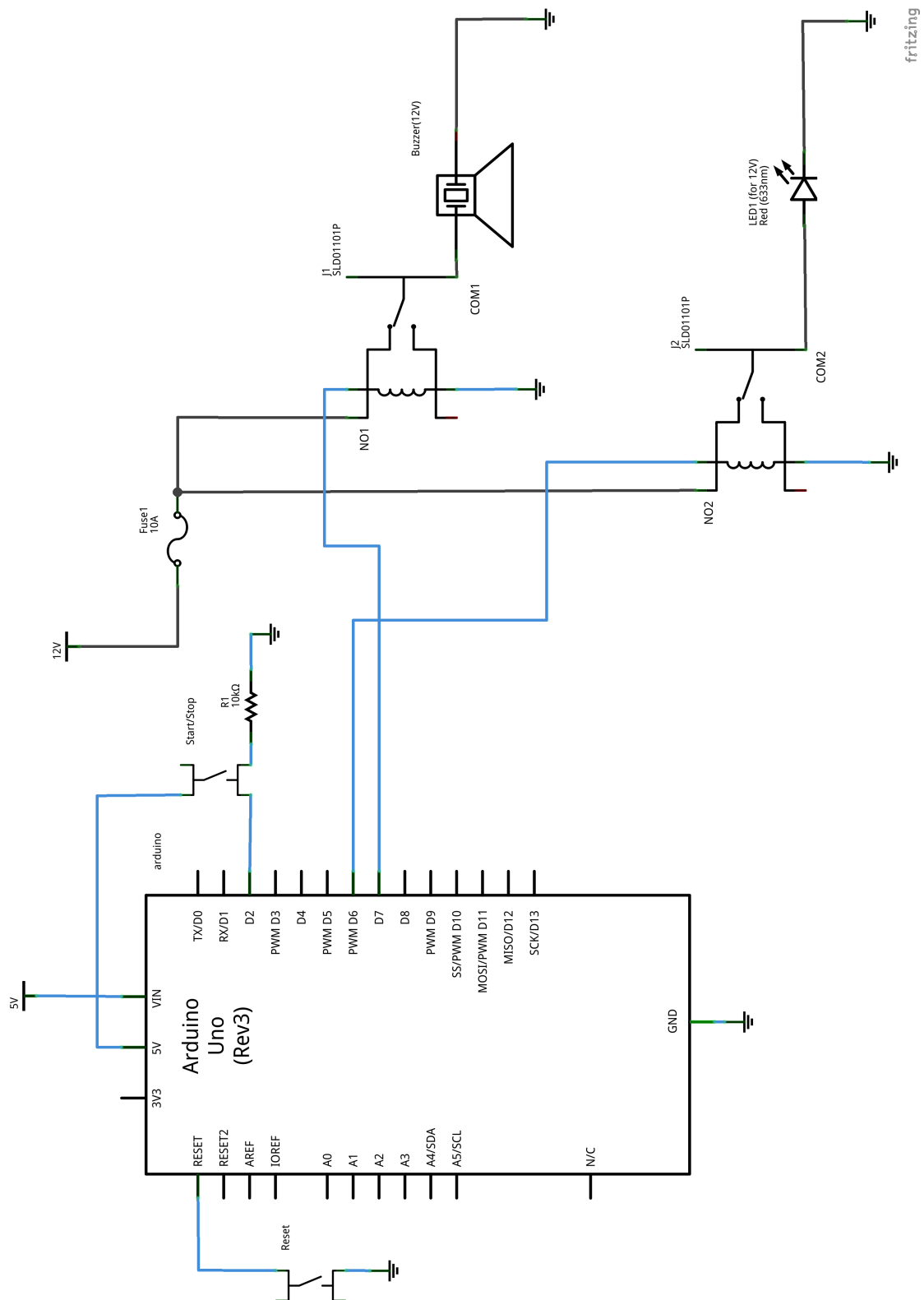
```
struct TIMER STAGE1_DEFAULT = {5, 2, 20} ; //ステージ 1(ゆっくり点滅) の LIMIT, STEP, TIME
struct TIMER STAGE2_DEFAULT = {2, 1, 10} ; //ステージ 2(早く点滅) の LIMIT, STEP, TIME
struct TIMER ENGAGE_DEFAULT = {1, 1, 5} ; //MCOM 起動の LIMIT, STEP, TIME
struct TIMER DISENGAGE_DEFAULT = {10, 1, 5} ; //MCOM 解除の LIMIT, STEP, TIME
```

ここで、*LIMIT* は長押し時間・各ステージの最低時間を指し、*STEP* は MCOM 解除毎にどれだけ *TIME*(後述)を増減するかを表しており、正の値だと各ステージ及び起動長押しは解除毎に *STEP* 秒短くなり、解除長押しは長くなります。*TIME* はそれぞれの長押し時間・各ステージの継続時間です。

2.2 組立について

2.2.1 基本的な組立作業

1. freemcom/freemcom.ino を arduino に書き込みます。
2. リレーシールド、ワイヤレスプロトシールドを arduino に接続します。
3. 下記回路図に従い、D2 ピンを起動/解除用押ボタンに接続します。(このスイッチはケース表面から押せる所に設置)
4. リレーシールドの NO2・COM2 を大音量ブザーに接続します。
5. リレーシールドの NO1・COM1 を赤色灯に接続します。
6. Arudino UNO の場合リセットピンを GND に落とすとリセットされるので、押ボタンスイッチを介して GND に落とせるようにする。このスイッチはケース内等、容易に押せない場所に設置してください。



2.2.2 遠隔監視を行う場合の追加の作業

1. 2つのXBeeのうち、Arduinoに接続するXBeeとPCに接続するXBee、PAN ID(通信可能なグループ番号)を決め、それぞれのシリアルナンバーを控えます。
2. XBee エクスプローラに「Arduinoに接続する方のXBee」を接続してから、XBee エクスプローラをPCに接続します。
3. X-CTU を起動し、通信相手のシリアルナンバーと PAN ID を書き込みます。
注意:シリーズ2では、PCに接続するXBeeのFunction Setを「Coordinator AT」に、Arduinoに接続するXBeeのFunction Setを「Router AT」に設定する必要があります。
4. XBee エクスプローラをPCから取り外してから、XBeeを取り外し、Arduinoに接続されたワイヤレスプロトシールドに接続します。
5. 上記2・3と同じ手順で、PCに接続するXBeeの設定を行います。

詳細はXBeeを日本国内で取り扱っている会社の一つである、スイッチサイエンスによる解説記事を参照して頂ければ、と思います。

XBee シリーズ1の設定解説 (http://mag.switch-science.com/2012/07/20/startup_xbee_s1)

XBee シリーズ2の設定解説 (http://mag.switch-science.com/2012/08/01/startup_xbee_zb)

第 III 部

ゲームでの使用方法

3 ゲームで使う

3.1 遠隔監視を使用しない場合の使用方法

1. ゲーム開始前に Arduino の電源を入れ、リセットボタンを押す。
2. 起動/解除ボタンを *ENGAGE.TIME* で指定した時間押し続けると、断続的にブザーが鳴り、LED が点滅を初めます。
3. *STAGE1.TIME* が経過すると *STAGE2.TIME* に入り、LED 点滅サイクルとブザーの断続間隔が短くなります。
4. 2～3 の間に起動/解除ボタンを *DISENGAGE.TIME* で指定された時間押し続けると、ブザーが短く鳴る&LED が 2 回点滅し、待機状態に戻ります。(この時、*STAGE1.TIME* は *STAGE1.STEP* だけ短くなり、同様に *STAGE2.TIME* も *STAGE2.STEP* だけ短くなります)
5. *STAGE2.TIME* が経過してしまうと、LED が点灯し、同時に 5 秒間ブザーが鳴り続けて「フリーズ」します。(これ以降、起動/解除ボタンを押しても初期状態には戻せません)
6. ゲームの勝敗が決する、あるいは時間切れでゲームが終了したら、本体のリセットボタンを押して初期状態に戻してください。

3.2 遠隔監視を行う場合の使用方法

1. ゲーム開始前に PC と XBee を接続し、Arduino の電源を入れる。
2. `monitor/monitor.py` を起動します。コマンドラインオプションでポートを指定することができます。また、`monitor.monitor.py` の *port_default* の値を書き換えたり、コマンドラインオプション入りのショートカットを作成することで、この手順を省く事ができます。
例: `./monitor.py COM3`
3. 「起動まで 5 秒長押し」と表示されたのを確認後、ゲームを開始してください。
4. 起動/解除ボタンを *ENGAGE.TIME* で指定した時間押し続けると、断続的にブザーが鳴り、LED が点滅を初めます。
5. *STAGE1.TIME* が経過すると *STAGE2.TIME* に入り、LED 点滅サイクルとブザーの断続間隔が短くなります。
6. 2～3 の間に起動/解除ボタンを *DISENGAGE.TIME* で指定された時間押し続けると、ブザーが短く鳴る&LED が 2 回点滅し、待機状態に戻ります。(この時、*STAGE1.TIME* は *STAGE1.STEP* だけ短くなり、同様に *STAGE2.TIME* も *STAGE2.STEP* だけ短くなります)
7. *STAGE2.TIME* が経過してしまうと、LED が点灯し、同時に 5 秒間ブザーが鳴り続けて「フリーズ」します。(これ以降、起動/解除ボタンを押しても初期状態には戻せません)
8. ゲームの勝敗が決する、あるいは時間切れでゲームが終了したら、本体のリセットボタンを押すか、監視ソフトのリセットボタンをクリックして初期状態に戻してください。

第 IV 部

トラブルシューティング

4 うまく動かない時は

4.1 遠隔監視を使用しない場合の確認事項

1. freemcom/freemcom.ino が正しく書き込まれているか確認する。書き込み時に何かエラーメッセージが出ている場合、書き込んでいません。
2. 配線が適切で、断線していないかテスターを使用して検査する。
 - (a) ピンがソケットに正しく差し込まれていて、間違っ隣りのピンに差し込んでいないかを確認する。
 - (b) シールドが正しく差し込まれているか。
3. 電源が供給されているか。Arduino とリレーシールドを動かすための電源と、LED 赤色灯やブザーを動かすための電源の両方が必要です。

4.2 遠隔監視を使用する場合の追加の確認事項

1. 正しいポートを指定しているか (未使用ポートならば、「〈ポート番号〉が見つかりません。」と表示されますが、不運にも何らかの機器がそのポートを使用していた場合、正常に表示されないでしょう)。
2. Arduino に正しくスケッチを書き込んでいるか (Arduino に USB ケーブル経由でのスケッチ書込時には、ワイヤレスプロトシールドの SERIAL SELECT スイッチを USB 側にしなければなりません)。
3. ゲーム時に、ワイヤレスプロトシールドの SERIAL SELECT スイッチが MICRO 側になっているか。
4. XBee の設定が正しいか。シリーズ 2 の場合 Coordinator と Router を取り違えていないか。

これらが全て正常でも通信の不具合が発生する場合、MCOM からの電波が PC まで届いていない可能性があります。その場合、以下の事をお試してください。

1. MCOM を PC に近づける
2. 遮蔽物を取り除く
3. XBee を U.FL コネクタ型に交換し外部アンテナを取り付ける

第 V 部

本プログラム及び使用ライブラリのライセンス、著作権

5 本プログラムの著作権及び使用ライセンス

- Copyright (c) 2014 AOKI Kiyohito, FreeMCOM for Airsoft develop squatron

本プログラムは上記が著作権を保有し、GNU GENERAL PUBLIC LICENSE Version 3 (or later) により提供されています。ライセンスの全文は COPYING を参照ください。

5.1 使用ライブラリの著作権及びライセンス

本プログラムは、以下のライブラリを使用しています。

- wxPython library
 - Copyright (c) 1992-2013 Julian Smart, Vadim Zeitlin, Stefan Csomor, Robert Roebling, and other members of the wxWidgets team, please see the acknowledgements section below.

Acknowledgements

The following is the list of the core, active developers of wxWidgets which keep it running and have provided an invaluable, extensive and high-quality amount of changes over the many of years of wxWidgets' life:

- Julian Smart
- Vadim Zeitlin
- Robert Roebling
- Robin Dunn
- Stefan Csomor
- Vaclav Slavik
- Paul Cornett
- Wlodzimierz 'ABX' Skiba
- Chris Elliott
- David Elliott
- Kevin Hock
- Stefan Neis
- Michael Wetherell

ライセンス原文は <https://www.wxwidgets.org/about/licence/> にあります。