

FreeMCOM for Airsoft v2.0.0 導入説明書

FreeMCOM for Airsoft development squatron

2015 年 4 月 9 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	セットアップに必要な部品一覧	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	6
2.1	規定値の調整について	6
2.2	組立について	6
第 III 部	ゲームでの使用方法	9
3	ゲームで使う	9
3.1	遠隔監視を使用しない場合の使用方法	9
3.2	遠隔監視を行う場合の使用方法	9
第 IV 部	トラブルシューティング	12
4	既知の制限とバグ	12
4.1	全ての環境で共通	12
4.2	Windows 環境のみ	12
5	うまく動かない時は	12
5.1	遠隔監視を使用しない場合の確認事項	12
5.2	遠隔監視を使用する場合の追加の確認事項	13
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	14
6	本プログラムの著作権及び使用ライセンス	14
6.1	FreeMCOM for Airsoft develop squatron メンバー紹介 (敬称略)	14
6.2	使用ライブラリの著作権及びライセンス	14
第 VI 部	参考文献等	16

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もしくはオブリタレーションに準じたルールで遊ぶための、Arduino を使用した M-COM/爆弾 (本説明書中では M-COM で統一) のクローンを製作するためのソフトウェアです。従って、ゲームの前に予め Arduino に本ソフトウェアを書き込み、ボタン・ブザー・赤色灯、スライドスイッチ及びキースイッチを接続しておく必要があります。

要するに、ラッシュモードでの利用法は、

- 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動し、破壊される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、万が一起動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。

要するに、オブリタレーションモードでの利用法は、

- 2 チームに分かれる。
- 各チームのフラッグの下に鍵を置く。紛失防止のために紐でフラッグに結わえるといいでしょう。
- フィールドの中央辺りに M-COM を置く。
- ゲームが開始されたら、M-COM を相手チームより先に確保したら、相手チームのフラッグまで運びます。
 - － 途中でヒットされたら、ヒットコールと共に M-COM をその場に置いてセーフティーゾーンに帰ってください。
- 無事相手チームのフラッグまで辿り着いたら、そこにある鍵でキースイッチの一方を「ON」に切り替えてから、ボタンを長押しして起動してください。もちろん、破壊されるまでは相手に解除されないようにする必要があります。
 - － もちろん、M-COM を起動されてしまった側は、解除する必要があります。(キースイッチの一方が「ON」のままだでも解除は可能です。)

第 II 部

セットアップ

1 セットアップに必要な部品一覧

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. 電源 (Arduino 用と、LED 赤色灯・ブザー用の 2 系統が必要)^{*1}
3. 大音量のブザー^{*2}
4. LED 赤色灯^{*3}
5. 押ボタンスイッチ (モーメンタリ) ×1 (起爆/解除用)
6. スライドスイッチ ×1 (ゲームルール切替用)
7. キースイッチ ×2 (オプリタレーションモードで使用。)
8. タクトスイッチ ×1 (リセットボタン用)
9. 10kΩ 抵抗 (プルダウン抵抗) ×3
10. ユニバーサル基板 ×1
11. リレーシールド (本書では SLD01101P を想定)
12. ケース^{*4}

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)

^{*1} Arduino 用の電源は、スマートフォン用のモバイルバッテリーを使うのが楽です。また、LED 赤色灯・ブザー用の電源は 12V の物を選択すると、安くて音量・輝度が大きい、自動車用の部品が使用可能になります。

^{*2} なお、とある開発メンバーはブザーの代わりに、自動車用のホーンを使用したら近所から苦情がきたそうです。

^{*3} 先述の開発メンバーは、12V 車対応のマーカーランプを使用しています。

^{*4} 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

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、`sketch/sketch.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 FUSE_DEFAULT = {1, 1, 5}; //MCOM 起爆の LIMIT, STEP, TIME
struct TIMER DEFUSE_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爆長押しは解除毎に *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

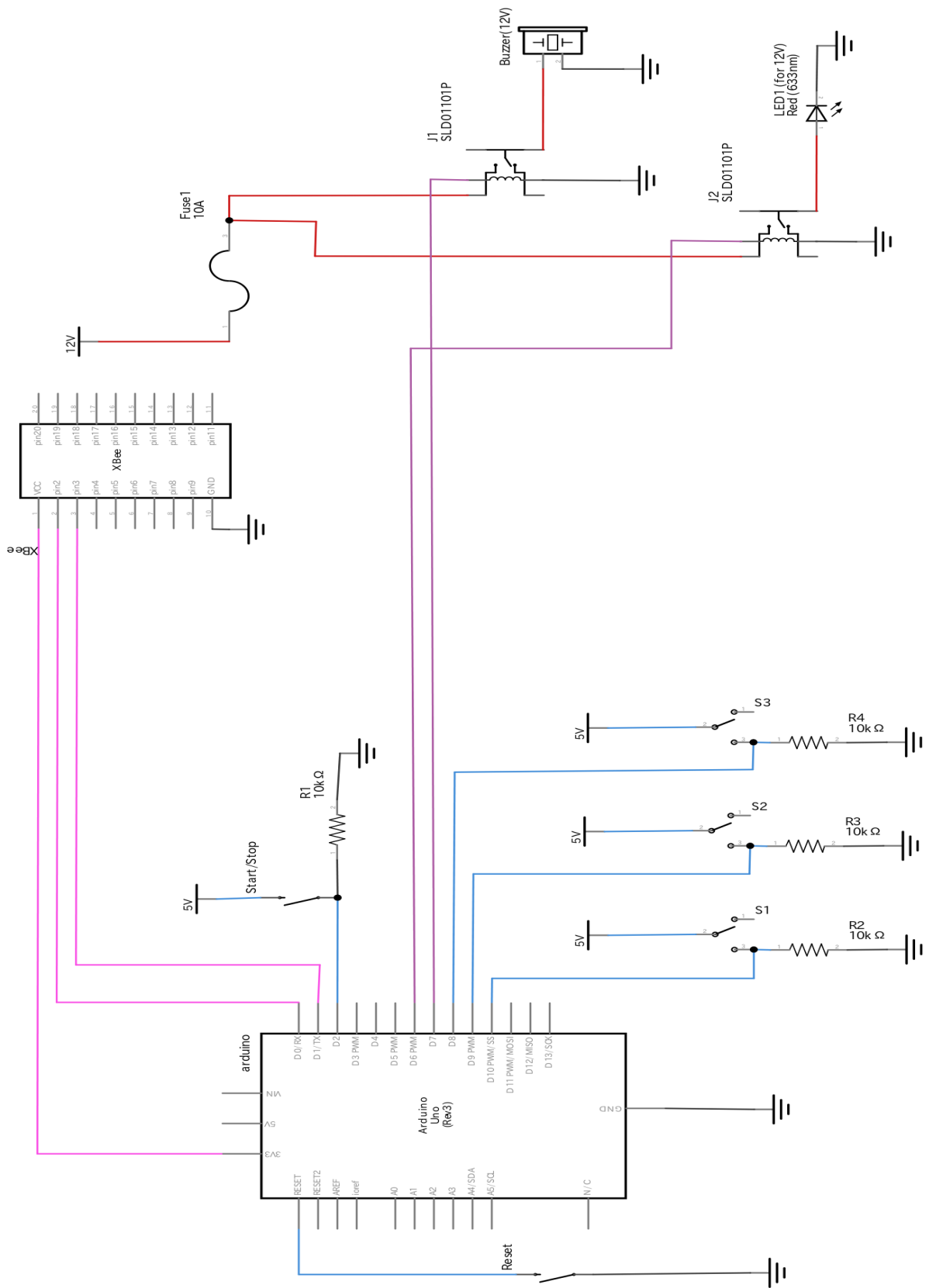
1. `sketch/sketch.ino` を `arduino` に書き込みます。
2. リレーシールド、ワイヤレスプロトシールドを `arduino` に接続します。
3. 下記回路図に従い、D2 ピンを起爆/解除用押ボタンに接続します。^{*5}
4. 同様に、D8・D9 ピンをキースイッチに接続します。^{*6}
5. 更に、D10 ピンをスライドスイッチに接続します。^{*7}
6. リレーシールドの NO2・COM2 を大音量ブザーに接続します。
7. リレーシールドの NO1・COM1 を赤色灯に接続します。
8. `Arudino UNO` の場合リセットピンを GND に落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。^{*8}

^{*5} このスイッチはケース表面から押せる所に設置します。

^{*6} ラッシュモードのみの場合、GND に落としてください。

^{*7} ラッシュモードのみの場合、やはり GND に落としてください。

^{*8} 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 の電源を入れる。この時、モード切り替えスイッチの設定が正しいかチェックしてください。^{*9}
2. オブリタレーションの場合は、相手フラッグの下に設置した鍵を使って、キースwitchの一方を ON にします。
3. 起爆/解除ボタンを *FUSE.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。
4. *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。
5. 2～3 の間に起爆/解除ボタンを *DEFUSE.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)
6. *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、起爆/解除ボタンを押しても初期状態には戻せません)
7. 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 を接続する。
2. Arduino の電源を入れる。
3. `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及び言語 (翻訳ファイルが存在する場合) を指定することができます。^{*10}

コマンドラインオプションは以下の 2 つが利用可能です。

`-p <ポート名>`

`-l <言語>`

COM3 を使用し、日本語表示する場合は、以下のようになります:

`$. /monitor.py -p COM3 -l ja`

4. ゲームモードが正しいか確認し、“起爆まで 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。

^{*9} 両方のキーを抜き取った状態でボタンを長押しして起爆カウントダウンを始めたら、ラッシュモードになっています。この方法での試験後はリセットを行ってください。

^{*10} `monitor.monitor.py` の `port_default` 及び `lang` の値を書き換える、あるいはコマンドラインオプション込みでショートカットを作成する、等の方法で楽をすることができます。

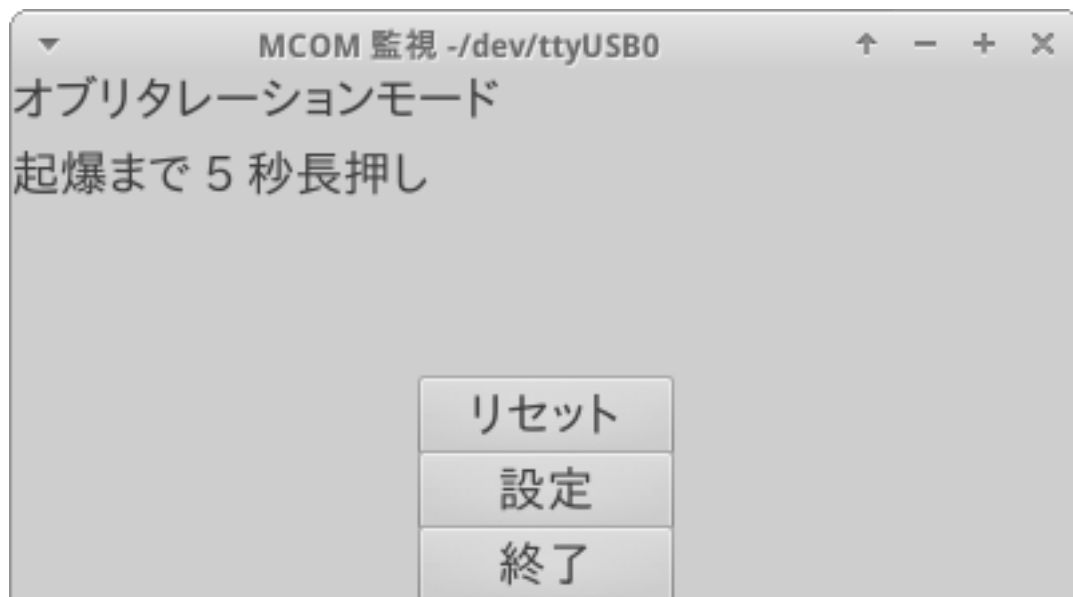


図 3.1 monitor.py を起動してゲーム開始準備が整った状態。ゲームモードが表示されている。

5. オブリタレーションの場合は、相手フラッグの下に設置した鍵を使って、キースイッチの一方を ON にします。
6. 起爆/解除ボタンを *FUSE.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。

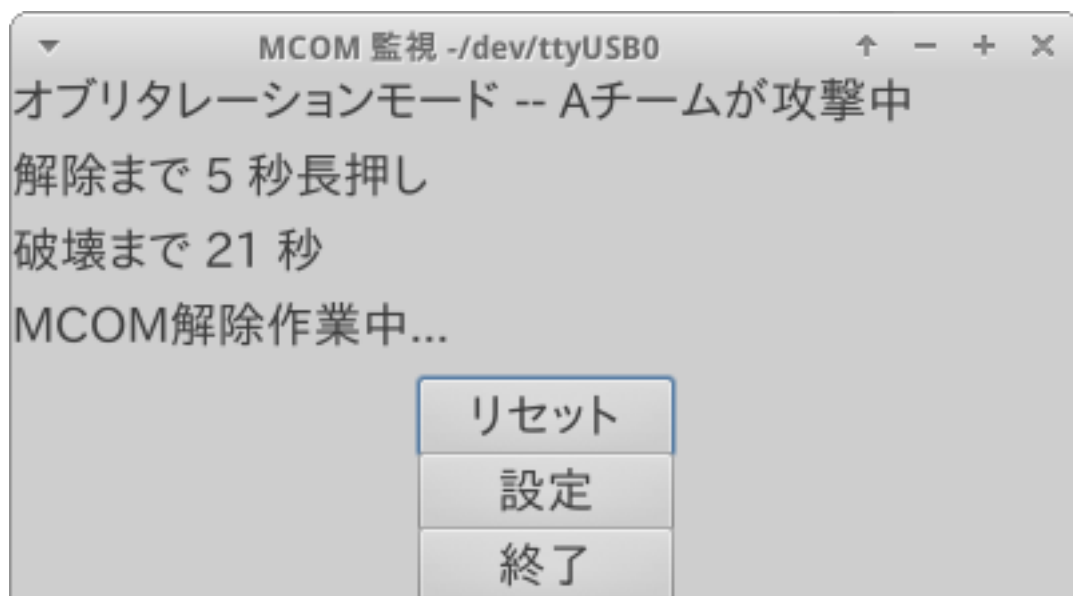


図 3.2 カウントダウン中。MCOM の起爆・解除作業中はこのように作業中の旨表示される。また、オブリタレーションモード中は、どちらが攻撃中かも表示される。

7. *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。

8. 6～7 の間に起爆/解除ボタンを *DEFUSE_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)
9. *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。^{*11}

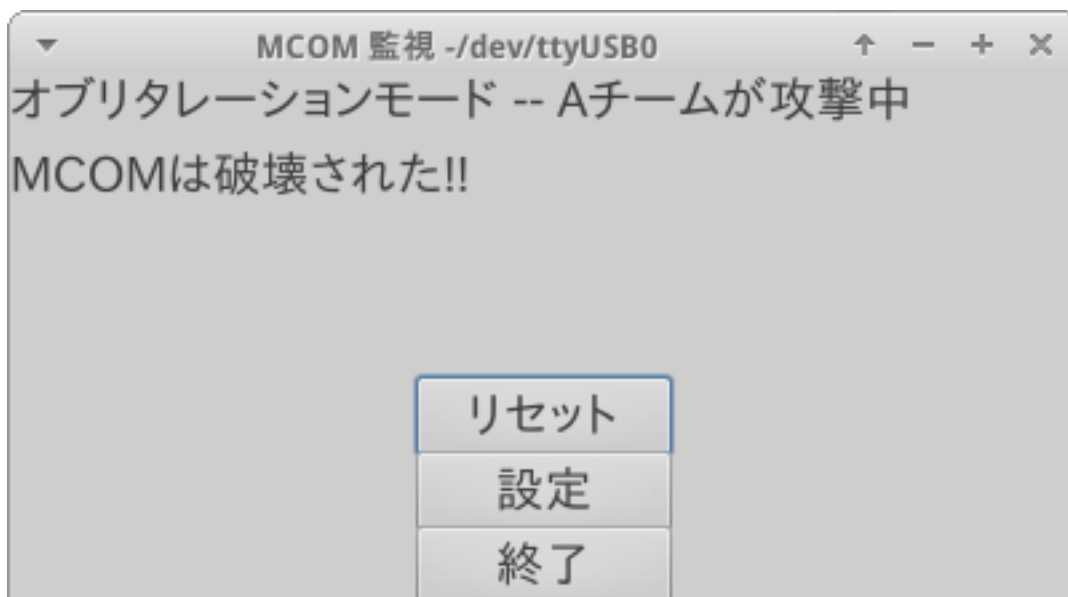


図 3.3 MCOM が破壊済の状態を表示

10. 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。

^{*11} これ以降、起爆/解除ボタンを押しても初期状態には戻せません。

第 IV 部

トラブルシューティング

4 既知の制限とバグ

4.1 全ての環境で共通

- MCOM を解除した直後、2 秒経過する前に再び長押しをすると、すぐに起爆準備状態になってしまう。
 - － 過剰な長押し対策として 2 秒のスリープを入れているが、`loop()` 関数に戻った直後に再びボタン長押し検出で引っかかって `mcom_stage1()` 関数に進んでしまう。

4.2 Windows 環境のみ

- 先に MCOM の電源が入の状態 XBee を PC に接続すると、カチカチと MCOM が暴走する。
- COM ポートを正しく選択していても、`monitor.py` が正常に認識しない。
 - － デバイスマネージャーで、「デバイスを削除」を実行後、一旦 XBee を PC から取り外して装着しなおすと改善される事がある。
 - － デバイスマネージャーで”マウスとそのほかのポインティング デバイス”を開き、”Microsoft Serial BallPoint”を”無効”に設定する。
 - － レジストリを操作する方法もあるようだが、当面は本書では触れない予定。(私の環境では、上記 2 つで改善されたため)

5 うまく動かない時は

5.1 遠隔監視を使用しない場合の確認事項

1. 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。
2. モード設定が正しいか。
3. `sketch/sketch.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。
4. 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、間違って隣のピンに差し込んでいないかを確認する。
特にプルダウン抵抗を挟んでアースに落とすピンが抜けていると、動作が非常に不安定になります。
 - (b) シールドが正しく差し込まれているか。

5.2 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)
 - (a) 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や、Windows 環境で「シリアルマウス」と検出されてしまっている場合は、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

6 本プログラムの著作権及び使用ライセンス

- Copyright (c) 2014-2015 FreeMCOM for Airsoft develop squatron <https://github.com/FreeMCOM>

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。

6.1 FreeMCOM for Airsoft develop squatron メンバー紹介 (敬称略)

- AOKI 'sambar' Kiyohito <https://sites.google.com/site/sambarsjunkyard/>

現在 AOKI Kiyohito 一名につき、開発メンバーを募集しています。

6.2 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。

第 VI 部

参考文献等

FreeMCOM for Airsoft の製作にあたり、参考にした書籍及び web サイト等を紹介いたします。

- Arduino をはじめよう 第二版 (<http://www.oreilly.co.jp/books/9784873115375/>)
- XBee で作るワイヤレスセンサーネットワーク (<http://www.oreilly.co.jp/books/9784873115306/>)
- Arduino 日本語リファレンス (<http://www.musashinodenpa.com/arduino/ref/>)
- スイッチサイエンスマガジン (<http://mag.switch-science.com/>)
- Python スタートブック (<http://gihyo.jp/book/2010/978-4-7741-4229-6>)
- [改訂第五版] L^AT_EX 2_ε 美文書作成入門 (<http://gihyo.jp/book/2010/978-4-7741-4319-4>) ^{*12}
- ものくろ ぼっくす (<http://www.monochrome-photo.info/>)

^{*12} 本文書執筆時点で、既に第 6 版が発行されています。これからお買い求めの方は <http://gihyo.jp/book/2013/978-4-7741-6045-0> を購入されることをお勧めします。