

小型ロボットに用いるマイクロコンピュータの放射線耐久実験 - 極限下でのレスキューロボットの開発に向けて -

大場 公隆* 武藤 啓太郎** 玉川 洋一** 平田 隆幸*

Damages tests of microcomputers under γ -rays. - Development of a rescue robot that works under an extremely severe situation -

Masataka OHBA*, Keitarou MUTOU**,
Youichi TAMAGAWA** and Takayuki HIRATA*

(Received January 30, 2008)

The damage tests of microprocessors were carried out under γ -rays. The PIC microprocessors produced by Microchip Technology Inc are inexpensive and popular CPU, so that they are used in autonomous small robots. Under γ -rays, we tested the reliability of the PIC12F675 and PIC16F84A that belonged to mid-range of the 8-bit PIC microprocessor's family. The source of γ -rays was Cesium 137. The source of γ -rays was mounted on the PIC directly. The PIC's plastic package was scraped by 0.7 mm in order to increase the exposure dose of the γ -rays to silicon chip in the plastic package of PIC. The LEDs connected to I/O Ports of the PIC were blinked by programs, which was monitored by video camera. Seven days radiation of γ -rays did not affect the function of PIC microprocessors.

Key words : Radioactive, Small autonomous robots, Microcomputer, Reliability, Damage, γ -rays

1. 緒言

惑星探査ロボットや災害救助ロボットの可能性として、自律分散型の群ロボットの研究が注目をあつめている。^[1] 自律分散型の群ロボットの特徴の一つとして、頑健さ (robustness) を挙げることができる。複数台のロボットが協力することで、複雑な仕事を達成する自律分散型の群ロボットは、アクシデントにより数台のロボットが故障したとしても大きな影響を受けずに、仕事を達成することができる。このシステムとしての頑健さは、惑星探査ロボットや災害救助ロボットに求められる特質の一つである。また、一般的に、群ロボットは小型ロ

ボットから構成されており、輸送が容易でありかつ通路が狭い状況においても機能するなど、惑星探査ロボットや災害救助ロボットに必要な条件を兼ね備えている。さらに、プログラムを変えるだけでさまざまな状況に対応できるので、惑星探査ロボットや災害救助ロボットとしての期待は大きい。

さて、群ロボットの大きな特徴の一つは、ある割合のロボットが故障しても仕事に差し支えないという頑健さをもっていることであるが、大部分のロボットが故障してしまうと達成すべき仕事をこなせなくなってしまう。群ロボットの使用が想定される惑星探査や深海底探査や災害救助においては、非常に過酷な状況が考えられる。例えば、惑星探査や宇宙空間での作業においては、高レベルの放射線が存在しているもとの作業が要求される。また、災害救助における大きなニーズの一つに、原子力発電所の事故現場での救助ロボットあるいは原子力発電システムへの修復ロボットの開発がある。原子力発電所の事故において、一番の問題点は言うまでもなく、放射

* 大学院工学研究科知能システム工学専攻

** 大学院工学研究科原子力・エネルギー安全工学専攻

* Human of Artificial Intelligent Systems Course,
Graduate School of Engineering

** Nuclear Power and Energy Safety Engineering
Course, Graduate School of Engineering

表1 放射線の種類と遮断材。

放射線	主な線源のエネルギー	遮蔽素材
α 線	約 5[MeV]	紙
β 線	0.018~2.27[MeV]	アルミ
γ 線	0.00591.17[MeV]	鉛, コンクリート
中性子	—	水

線の存在である。それゆえ、宇宙空間や原子力発電所の事故現場などの高レベルの放射線が存在する極限状況下でも、動作する小型ロボットの開発が必要とされる。

高レベルの放射線下で作動する小型ロボットの開発においては、ロボット自体の被爆の問題もあるが、自律分散型の知能をになう CPU が放射線により誤動作あるいは破壊されて、動かなくなる点が一番大きな問題である。

ここでは、放射線下での CPU の耐久性を調べた。Microchip Technology 社の PIC マイクロコンピュータは小型でコストパフォーマンスが良いため、群ロボットの CPU として用いられることが多いマイクロコンピュータである。そこで、PIC マイクロコンピュータの中でも広く使用されている 16F84A と 12F675 の γ 線への耐久性を調べた。

2. 放射線と回路素子

放射線には、 α 、 β 、 γ 、X、陽子、中性子線がある^{[2][3]}。 α 線は He の裸の原子核、 β 線は電子、 γ 線と X 線は電磁波、陽子線は陽子、中性子線は中性子である。 γ 線と X 線は発生のしかたが異なり、一般に γ 線のほうが高エネルギーである。 γ 線は原子核の β 崩壊後に原子核から発生し、X 線は外部からエネルギーを吸収し励起した電子から発生する。また、放射線は電荷を持つ荷電粒子と電荷を持たない非荷電粒子に分類できる。 α 、 β 、陽子線は荷電粒子に、 γ 、X、中性子線は非荷電粒子に分類される。荷電粒子は原子からクーロン力をうけるため透過力は低い。表1に、 α 、 β 、 γ 、中性子線の代表的なエネルギー、一般に遮蔽に用いられる物質をまとめておく。ここでは透過力が強かつ放射線としての取り扱いが比較的容易な γ 線に注目して、耐久試験をおこなった。

γ 線は放射性物質から得ることができる。放射性物質は自発的な核反応を起こす物質であり、放射線の種類によって α 、 β 、 γ 線源に分類される。本研究では、 γ 線源としてセシウム 137 ($^{137}_{55}\text{Cs}$) (0.662MeV) をもちいた。セシウム 137 はウランの核分裂によって得られる。ウラ

ン (U) が核分裂を起こすと、質量数が 90~100 の原子核と 135~145 の原子核の 2 つに分裂する場合が多い。これらの原子には中性子が多く不安定である、そこで遅発中性子の放出や中性子が陽子と電子に分裂し電子を放出する β 崩壊を行うことで安定しようとする。核分裂片にテルル 137 が生じると、 β 崩壊を繰り返しヨウ素 137、ゼノン 137、セシウム 137 の順に変化する^[4]。

2.1 放射線による集積回路へのダメージ

放射線による回路素子のダメージを考える。ガンマ線と物質の相互作用としてトムソン散乱、光電効果、コンプトン効果、電子対生成、光核反応が知られている。トムソン散乱は電磁波が物質の軌道電子にあたり方向を変える現象である。このとき電磁波はエネルギーを失わず、電子は影響を受けない。光電効果は光電吸収とも呼ばれ電磁波が原子に吸収される。吸収された電磁波のエネルギーは軌道電子にうけわたされ軌道を外れる。コンプトン効果は電磁波と軌道電子との相互作用である。軌道電子は電磁波のエネルギーの一部を受け取り原子の軌道から外れる。一方、電磁波はエネルギーを失った分波長が長くなり散乱される。電子対生成は高エネルギーの電磁波が物質中で一対の電子と陽電子に変化する現象である。この現象には陰陽電子 2 個分のエネルギー 1.02[MeV] 以上が必要である。さらに 10[MeV] の強力な電磁波は原子核の団結を断ち切る光核反応を引き起こす。

今回使用した線源ではトムソン散乱、光電効果、コンプトン効果が集積回路の内で発生することが考えられる。集積回路では 1 次的なデータの保存に RAM を使用する。RAM は情報の保存を電荷の蓄積で行っている。光電効果、コンプトン効果の結果、電子が放出されると RAM などの bit 化けを起こす可能性がある。

3. 放射線源の較正実験

実験に使用した線源の放射能の強さを正確に求めるために、放射能線源の較正実験をおこなった。放射線の強さが分かっている基準線源と比較することで放射線源の放射能の強さを求めた。具体的には、線源から一定距離に γ 線観測装置を設置し、対象線源と基準線源で同一時間カウントし、その比率から放射能の強さを計算した。図1に実験の模式図と実物の写真を示す。線源とシンチレータを 30cm 離して設置し 600 秒間カウントした。シンチレータは蛍光体封入管、光電子増倍管 (PMT)、多

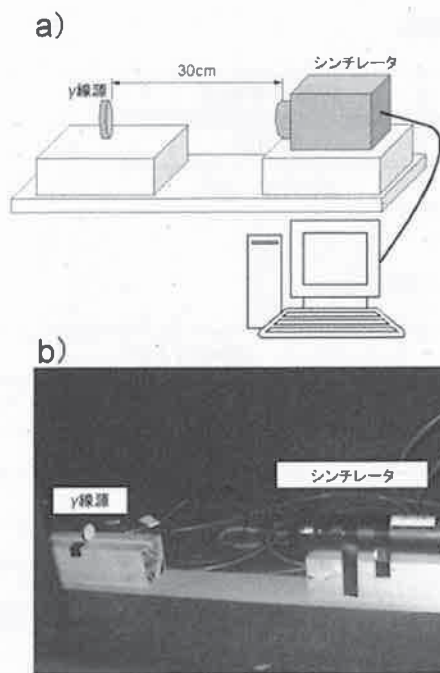


図1 放射線源の校正実験. a) 模式図, b) 実験装置の写真.

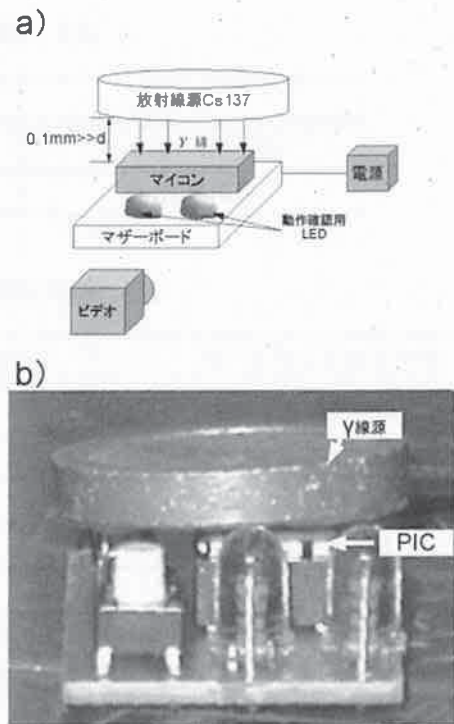


図2 γ 線照射実験. a) 模式図, b) 実験装置の写真.

重パルス高分析器から構成され、 γ 線が入射すると蛍光体原子が励起される。蛍光体は吸収したエネルギーに応じて光を発する。この光は光電子増倍管で電気信号に変えられ、多重パルス高分析器に送られ、放射線量がカウントされる。実験は外光が入らないように暗室でおこなった。使用した放射能線源の強さは 2.3173×10^5 [Bq] であった。

4. マイクロコンピュータ (PIC) への照射実験

マイクロコンピュータとしては、PIC (Peripheral Interface Controller) とよばれる CPU をもちいた^[5]。今回使用した PIC は、Microchip Technology 社製の 8 ビットマイクロコンピュータの一種であり、主に周辺機器をコントロールするマイクロコントローラーに分類され、小型ロボットなどの制御にも使われる。PIC には、ベースラインシリーズ (命令長 12 ビット)、ミッドレンジシリーズ (命令長 14 ビット)、ハイエンドシリーズ (命令長 16 ビット) など多様なレンジのマイクロコンピュータが存在する。(さらに 16 ビット 32 ビットの物も存在する。) ここでは、ミッドレンジシリーズに属する PIC12F675 と PIC16F84A を用いた^{[6][7]}。PIC は、演算処理ユニットのみならず、プログラムメモリ、データメモリ、入出力ポート (I/O Ports) などを内蔵しており、電源を供給するだけで動作する。表 3 に今回もちいた PIC の性能を

まとめておく。

γ 線を照射したとき、PIC が正常に動作するかをモニターする実験をおこなった。PIC が正常に動作しているかどうかは、PIC に LED を接続し、LED を一定間隔で順番に点滅させるプログラムを走らせ、LED の点滅をビデオでモニターする方法で判断した。また、 γ 線の照射は線源を PIC の上に直接置いておこなった。(図 2 を参照)

パッケージの表面を 0.7mm 削った PIC を照射実験に用いた。これは、パッケージを削ることにより、トランジスタ部分により多くの γ 線を照射できるようにするためである。削る量は、パッケージ内の配線を傷つけずかつ、最大になるように決定した。

LED を点滅させるために使用した PIC マイコン (PIC12F675, PIC16F84A) 用マザーボードの回路をそれぞれ図 3, 図 4 に示す。この回路は PIC マイコン、振動子 (セラロック 20MHz)、LED から構成される。回路は接触不良による誤動作をなくするためにプリント基板で作成した。パーツをマザーボードに実装した実験用回路 (12F675, 16F84A) の写真をそれぞれ図 5, 図 6 に示す。

4つの LED を順番に点滅させるプログラムは、C 言語を用いてプログラミングした。動作確認用のプログラムを付録に示す。コンパイラは PICCLite バージョン 9.50PL2 を使用した。PICCLite は HI-TECH Software

表2 実験に用いた放射線源の放射能.

	基準線源	計測対象線源
パルスカウント数	28474	21162
放射能 [Bq]	$3.118 * 10^5$	$2.3173 * 10^5$

表3 実験に使用したマイクロコンピュータの性能.

マイクロコンピュータ	16F84A-20/P*	12F675-1/P*
最大動作周波数 (MHz)	20	20
フラッシュメモリ (ワード)	1024	1024
データメモリ (byte)	68	64
動作電圧 (V)	2~6	2~5.5
動作温度 (°C)	0~70	-40~85

*20/Pは20MHzまでの動作が確認されたもので他に04P(4MHzまで動作)などがある。Iは温度レンジを示す(-40~85°C, -のときは0~70)。またPはPDIP(プラスチックパッケージ)を意味している。

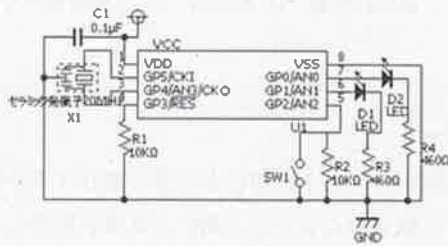


図3 12F675用のマザーボードの回路図.

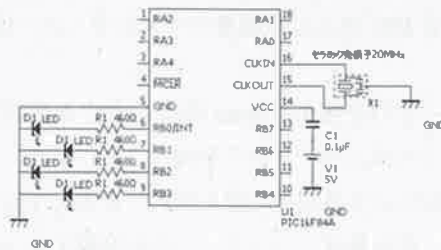


図4 16F84Aのマザーボードの回路図.

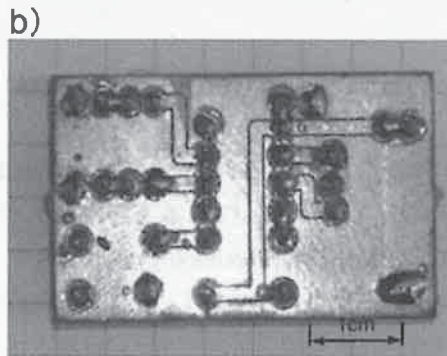
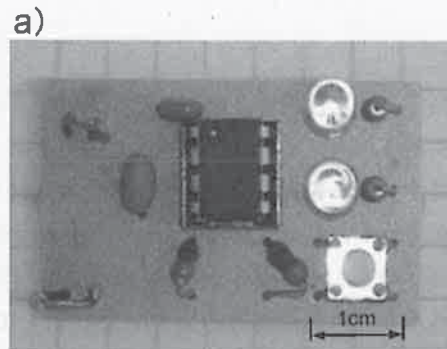


図5 PIC12F675用マザーボード、a)部品を実装した面、b)パターン面.

社のC言語用コンパイラであり、これは商用のPICCの機能を限定し、無料で使用できるようにしたものである。なおPIC12F675とPIC16F84Aに関しては制限なしで使える。実行ファイルはPIC12F675が799バイト、PIC16F84Aが1079バイトであった。

γ線をPIC12F675、PIC16F84Aにそれぞれに連続1週間照射して、4つのLEDが順番に点滅しなくなる誤動作が生じないかを調べた。プログラムによって、LEDを連続点滅させているので回路素子の一部にダメージを

受けると、プログラムの暴走によりLEDの点滅が見られなくなると考えられる。ここでは、4つのLEDが順番に点滅しているときは、PICはダメージを受けていないとみなしている。さらに、回路が正しく動作しているかを正確にチェックするため、初日と最終日を含む5日分のビデオ撮影した3分間の動画を詳細に調べた。実

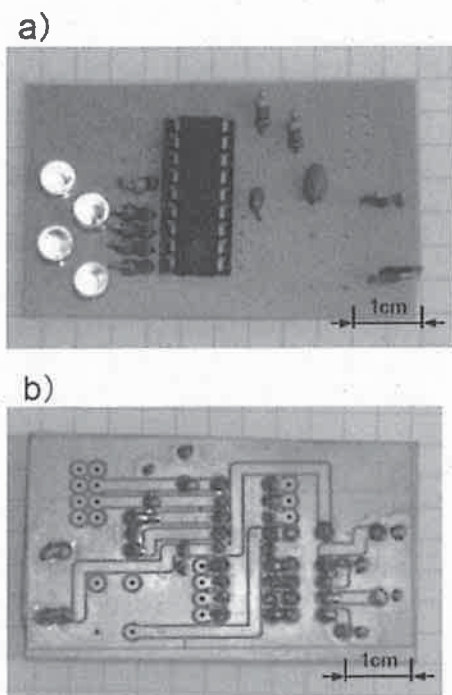


図6 PIC16F84A用マザーボード。a) 部品を実装した面、b) パターン面。

験を記録したビデオを解析したが順番の逆転等の誤動作は確認できなかった。

5. まとめと展望

ロボットに搭載するマイクロコンピュータに放射線をあて動作を観察した。マイクロコンピュータはMicrochip Technology社のPIC12F675I/PとPIC16F84A-20/Pを使用した。マイクロコンピュータの動作を確認するための簡単な回路とプログラムを作成した。回路にはLEDを搭載し、これを順に点灯させることでマイクロコンピュータの動作確認をおこなった。実験に使用した γ 線源は較正実験を行い放射能の強さを測定した。測定はあらかじめ放射能の強さがわかっている別の線源と比較する方法をとった。実験に使用した放射線源は $6.26303905[\mu\text{Ci}]$ であった。この実験ではマイクロコンピュータの故障および誤動作は確認できなかった。

誤動作がみられなかった要因として1)放射線の出力が足りなかった2)マイクロコンピュータの全ての機能が正常に動作しているかをチェックするプログラムになっていないなどが考えられる。1)について、今回使用した放射線源は $6.26[\mu\text{Ci}]$ で胸部レントゲンの $0.1[\text{mCi}]$ と比べて小さい。2)について、今回の実験ではマイクロコンピュータの単純な動作について調べており、加算

演算、ビットシフト演算を行いIOポートに出力するというものであったため、使用したメモリが少なくビット化けがあっても正常に動作しているようにみえる可能性がある。今後マイクロコンピュータの全ての機能を使用するようにプログラムにも改良の余地がある。

救助ロボットが活動する場面は人間が立ち入れない極限状態が想定される。今回の実験では、限られた実験条件ではあるが極限状態の一種である放射線照射下でロボットの頭脳にあたるCPUが正常に動作することを確認できた。しかし今回の実験は、特定のマイクロコンピュータへの γ 線照射実験であった。PICは集積度がそれほど高くないので、より集積度が高く誤動作を起こしやすいCPU(例えばIntel Core2 Duo)や γ 線以外の放射線によるマイクロコンピュータの誤動作を調べる実験をおこないたい。また、今回はマイクロコンピュータに直接 γ 線を照射する実験であったが、より現実的な状況(救助ロボットへCPUを搭載した状況)での放射線下でのロボットへの影響を調べることも必要であろう。さらに、惑星探査ロボットなどを視野に入れた場合、振動や温度・湿度によるロボットへの影響を調べることも重要である。今後、様々な極限状況下で活躍できる群ロボットの開発のための基礎データを収集したいと考えている。

謝辞

較正実験を手伝っていただいた前川祐希君、さまざまな議論をしてくださった西川嗣雄教授、田中ダン講師、知能システム工学科非線形科学研究室、原子力工学専攻放射線粒子物理実験室のメンバー皆様に感謝します。

参考文献

- [1] F. Mondada, L. M. Gambardella, D. Floreano, S. Nolfi, J.-L. Deneubourg, and M. Dorigo: IEEE Robotics & Automation Magazine, 12-2,21 (2005).
- [2] 安西育郎:図解雑学放射線と放射能, ナツメ社 (2007)
- [3] 竹田敏一:図解雑学原子力, ナツメ社 (2002).
- [4] 日本アイソトープ協会: アイソトープ便覧, 丸善 (1984).
- [5] 後関哲也:電子工作のためのPIC活用ガイドブック, 技術評論社 10-13(2000).
- [6] Microchip Technology:PIC12F629/675 Data Sheet, Microchip Technology (2003).

[7] Microchip Technology: PIC16F84A Data Sheet, Microchip Technology (2001).



Faint, illegible text in the left column, likely bleed-through from the reverse side of the page.

Faint, illegible text in the left column, likely bleed-through from the reverse side of the page.

Faint, illegible text in the left column, likely bleed-through from the reverse side of the page.

Faint, illegible text in the left column, likely bleed-through from the reverse side of the page.

Faint, illegible text in the left column, likely bleed-through from the reverse side of the page.

Faint, illegible text in the right column, likely bleed-through from the reverse side of the page.

Faint, illegible text in the right column, likely bleed-through from the reverse side of the page.

Faint, illegible text in the right column, likely bleed-through from the reverse side of the page.

Faint, illegible text in the right column, likely bleed-through from the reverse side of the page.

Faint, illegible text in the right column, likely bleed-through from the reverse side of the page.

付録1 PIC12F675 動作プログラム

```

#include<pic.h>
#include"delay.h"
__CONFIG(UNPROTECT & BOREN & MCLRDIS & PWRTEN & WDTDIS & INTIO);
__IDLOC(0100);
ioport(){
    TRIS0=0;
    TRIS1=0;
    ANSEL=0x00; /*all is digital_IO*/
}
main(){
    char i=0;
    ioport();
    while(1){
        for(i=0;i<4;i++){
            GPIO=i;
            DelayMs(500);
            DelayMs(500);
            GPIO=0;
            DelayMs(500);
        }
    }
}

```

付録2 PIC16F84A 動作プログラム

```

#include<pic.h>
#define DILAY 25000000
__CONFIG(UNPROTECT & PWRTEN & WDTDIS & HS);
main(){
    int i,j;
    TRISA=0xff;
    TRISB=0x00;
    while(1){
        PORTB=0x01;
        for(i=DILAY;i>0;--i);
        for(i=DILAY;i>0;--i);
        for(i=DILAY;i>0;--i);
        for(i=DILAY;i>0;--i);
        for(j=0;j<4;++j){
            PORTB<<=1;
            for(i=DILAY;i>0;--i);
            for(i=DILAY;i>0;--i);
            for(i=DILAY;i>0;--i);
            for(i=DILAY;i>0;--i);
        }
    }
}

```

