

ML62Q1000 シリーズ
**ROM 未使用領域アクセスリセット機能
の使用方法**

初版 発行日 2021 年 3 月 24 日

ご注意

- 1) 本資料の記載内容は改良などのため予告なく変更することがあります。
- 2) 本製品をご使用の際は、最新の製品情報をご確認の上、絶対最大定格、動作条件その他の指定条件の範囲内でお使いください。指定条件の範囲を超えて使用された場合や、使用上の注意を守ることなく使用された場合、その後に発生した故障、誤動作等の不具合、事故、損害等については、ラピステクノロジー株式会社（以下、「当社」といいます）はいかなる責任も負いません。また、指定条件の範囲内のご使用であっても、半導体製品は種々の要因で故障・誤作動する可能性があります。万が一製品が故障・誤作動した場合でも、その影響により人身事故、火災損害等が起こらないよう、お客様の責任において、ディレーティング、冗長設計、延焼防止、バックアップ、フェイルセーフ等お客様の機器・システムとしての安全確保を行ってください。
- 3) 本資料に記載されております応用回路例やその定数、ソフトウェア等の情報は、半導体製品の標準的な動作例や応用例を説明するものです。お客様の機器やシステムの設計においてこれらの情報を使用する場合には、お客様の責任において行ってください。また、量産設計をされる場合には、外部諸条件を考慮していただきますようお願いいたします。これらのご使用に起因して生じた損害等に関し、当社は一切その責任を負いません。
- 4) 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の技術情報は、それをもって当該技術情報に関する当社または第三者の知的財産権その他の権利を許諾するものではありません。したがって、当該技術情報を使用されたことによる第三者の知的財産権に対する侵害またはこれらに関する紛争について、当社は何ら責任を負うものではありません。
- 5) 本製品は、一般的な電子機器（AV機器、OA機器、通信機器、家電製品、アミューズメント機器など）および本資料に明示した用途へのご使用を意図しています。
本製品を、特に高い信頼性が要求される機器（車載・船舶・鉄道等の輸送機器、幹線用通信機器、交通信号機器、防災・防犯装置、安全確保のための装置、医療機器、サーバー、太陽電池、送電システム等）に使用される際は、必ず当社へご連絡の上、書面にて承諾を得てください。
当社の意図していない用途に製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
また、本製品は直接生命・身体に危害を及ぼす可能性のある機器・システム、極めて高い信頼性を要求される機器（航空宇宙機器、原子力制御機器、海底中継機器等）には、使用できません。
- 6) 本資料に掲載されております製品は、耐放射線設計がなされておられません。
- 7) 本資料に記載されております情報は、正確を期すため慎重に作成したのですが、万が一、当該情報の誤り・誤植に起因する損害がお客様に生じた場合においても、当社はその責任を負うものではありません。
- 8) 本製品のご使用に際しては、RoHS 指令など適用される環境関連法令を遵守の上ご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は一切の責任を負いません。
- 9) 本製品および本資料に記載の技術を輸出または国外へ提供する際には、「外国為替及び外国貿易法」、「米国輸出管理規則」など適用される輸出関連法令を遵守し、それらの定めにしたがって必要な手続を行ってください。
- 10) 本資料に記載されている内容または本製品についてご不明な点がございましたらセールスオフィスまでお問い合わせください。
- 11) 本資料の一部または全部を当社の許可なく、転載・複写することを堅くお断りします。

Copyright 2021 LAPIS Technology Co., Ltd.

ラピステクノロジー株式会社

〒222-8575 神奈川県横浜市港北区新横浜 2-4-8

<https://www.lapis-tech.com/>

目次

1. はじめに	1
2. WDT リセットの代替案として利用可能なリセット機能	2
3. 変更方法	3
3.1. スタートアップファイル中のブレーク割込みハンドラの変更	3
3.2. スタートアップファイル中のプログラム開始時処理(\$\$BEGIN)の変更	4
3.3. スタートアップファイル中のコードオプションの変更	4
3.4. プログラム中のエラー処理の変更	5
4. 確認方法	6
4.1. オンチップデバッグ接続時の確認方法	6
4.2. オンチップデバッグ未接続時の確認方法	6
4.2.1. プログラム例	7
4.2.2. ROM 未使用領域リセットチェック処理	10
4.2.3. ブレークリセットチェック処理	11
4.2.4. ビルド・動作確認	12

1. はじめに

ML62Q1000 シリーズのマイコンでは、何らかの異常発生時の対処として、無限ループにより、ウォッチドッグタイマ (WDT) リセットを発生させることを推奨しています。

しかし、システムによっては、WDT を使わない場合もあり、その場合の代替案として ROM 未使用領域アクセスリセット機能を使用する方法を本書に示します。

ここに記載する内容は、あくまでも一つの代替案であり、システムによっては適用できない場合もありますので、十分な検討・評価をお願いします。

2. WDT リセットの代替案として利用可能なリセット機能

何らかの異常発生時の対処として、WDT リセットを発生させることを推奨しているのは、以下の理由によります。

- ・ソフトウェアでリセットを発生させることができる
- ・CPU だけでなく、周辺回路もリセットできる

WDT リセットの代替案としては、上記 2 点を満足するリセット機能が妥当と考えられます。ML62Q1000 シリーズでは、WDT リセット以外に上記 2 点を満足するリセット機能としては、ROM 未使用領域アクセスリセット機能があります。

条件として、ROM 未使用領域アクセスリセット機能を使うには、コードオプションによりこの機能を許可することが必要です（初期値は許可）。

ROM 未使用領域アクセスリセット機能の詳細については『ML62Q1000 シリーズ ユーザーズマニュアル』の「29.3.2 ROM 未使用領域アクセスリセット機能」を、コードオプションの詳細については同ユーザーズマニュアルの「26. コードオプション」を参照してください。

3. 変更方法

3.1. スタートアップファイル中のブレーク割込みハンドラの変更

\$\$brk_reset は brk 命令 (=0xFFFF) を実行したときの分岐先となるブレーク割込みハンドラのラベルになります。

無限ループにより WDT リセットを待つ箇所を、ROM 未使用領域への分岐命令に置き換えます。

ROM 未使用領域については、『ML62Q1000 シリーズユーザーズマニュアル』の「29.3.2 ROM 未使用領域アクセスリセット機能」を参照してください。

【変更前】

```

$$start_up:
    tb      INITE
    bz      $begin

$$brk_reset:
$$error:
    mov     r0,    #00h
    st      r0,    WDTMOD
    mov     psw,   #02h

;-----
;   reset SFRs
;   If you don't need the below then remove it.
;
;   ...
    st     er0,   BRECON3
;-----
    bal    $          ; wait for Watch Dog Timer reset
$begin:
    ...

```

【変更例:スモールモデルの場合】

```

$$start_up:
    tb      INITE
    bz      $begin

$$brk_reset:                ; ブレーク割込み処理の名前 (ラベル)
$$error:
    b       0FFFFh         ; 0FFFFh 番地 (ROM 未使用領域) へ分岐させることにより
                          ; ROM 未使用領域アクセスリセットを発生させます

$begin:                    ; 通常の処理の開始を示すラベル
    ...

```

【変更例:ラージモデルの場合】

```

$$start_up:
    tb      INITE
    bz      $begin

$$brk_reset:                ; ブレーク割込み処理の名前 (ラベル)
$$error:
    b       7:0FFFFh      ; 7:0FFFFh 番地 (ROM 未使用領域) へ分岐させることにより
                          ; ROM 未使用領域アクセスリセットを発生させます

$begin:                    ; 通常の処理の開始を示すラベル
    ....

```

3.2. スタートアップファイル中のプログラム開始時処理 (\$\$begin) の変更

ROM 未使用領域アクセスリセットが発生すると SRSTAT レジスタの FIAR フラグが 1 にセットされます。リセット後のプログラム開始処理 (\$\$begin) にて、FIAR フラグをチェックし、FIAR フラグが 1 にセットされていたら、FIAR フラグに 1 を書き込んで FIAR フラグをクリアしてください。

【変更例】

```

$begin:
    tb      FIAR      ; 通常の処理の開始を示すラベル
                ; FIAR フラグ (SRSTAT の 0 ビット目) をチェック
                ; (ROM 未使用領域アクセスリセット発生の有無チェック)
    bz      $normal  ; FIAR フラグが 0 ならば、$normal へ分岐
                ; FIAR フラグが 1 の場合、必要に応じて何らかの処理を記述
                ; (この例では以下のみ)
    sb      FIAR      ; FIAR フラグに 1 を書き込んで FIAR フラグをリセット
$normal:
    ....

```

3.3. スタートアップファイル中のコードオプションの変更

コードオプションにて、ROM 未使用領域アクセスリセットを許可 (ビット 12 を 1 (初期値))、WDT を動作禁止 (ビット 0 を 0) に変更します。

【変更前】

```

;-----
;      Setting the code-option data (for ML621XXX)
;-----
    ...

    cseg #?? at 0?fd0h      ; address(?はお使いのマイコンによって異なります)
    dw      0fffdh         ; 0?fd0h, PCERMD=1, WDTMD=1
    ...

```

【変更例】

```

;-----
;      Setting the code-option data (for ML621XXX)
;-----
    ...

    cseg #?? at 0?fd0h      ; address(?はお使いのマイコンによって異なります)
    dw      0fffch         ; 0?fd0h, PCERMD=1, WDTMD=0
    ...

```

3.4. プログラム中のエラー処理の変更

プログラム中の処理において、何らかの異常を検知した際に、無限ループにより WDT リセットを待つ箇所を、ブレーク命令に置き換えます。

【変更前】

```
void error_proc( void )
{
    .....
    for(;;){
        /* WDT リセット発生待ち */
    }
}
```

【変更例】

```
void error_proc( void )
{
    .....
    __asm("brk"); /* ブレーク命令 */
}
}
```


4. 確認方法

オンチップデバッグ接続時には ROM 未使用領域アクセス時に FIAR フラグは 1 にセットされますが、リセットは発生しません。
このため、オンチップデバッグ接続時とオンチップデバッグ未接続時の確認方法が異なります。

4.1. オンチップデバッグ接続時の確認方法

【オンチップデバッグ接続時の動作】

オンチップデバッグ接続時には、ブレーク割込みハンドラへ分岐後、その中で未使用領域へ分岐し 0xFFFF (=brk 命令) を実行するため、最終的にはブレークリセットが発生し、CPU のみリセットという動作になります。

【確認方法】

ROM 未使用領域へ分岐するコードを実行し、SRSTAT レジスタの FIAR フラグが 1 にセットされることを確認してください。

ROM 未使用領域へ分岐するコードを実行し、SRSTAT レジスタの FIAR フラグが 1 にセットされない場合、コードオプションが正常に設定されていないことが想定されます。コードオプションにて、ROM 未使用領域アクセスリセット機能が有効になっていることをご確認ください。

4.2. オンチップデバッグ未接続時の確認方法

【オンチップデバッグ未接続時の動作】

オンチップデバッグ未接続時には、ROM 未使用領域に分岐し、プログラムカウンタ (PC) が領域外のプログラムを実行しようとしたことを検知した時点で ROM 未使用領域アクセスリセットが発生します。
このため、ブレークリセットは発生しません。

【確認方法】

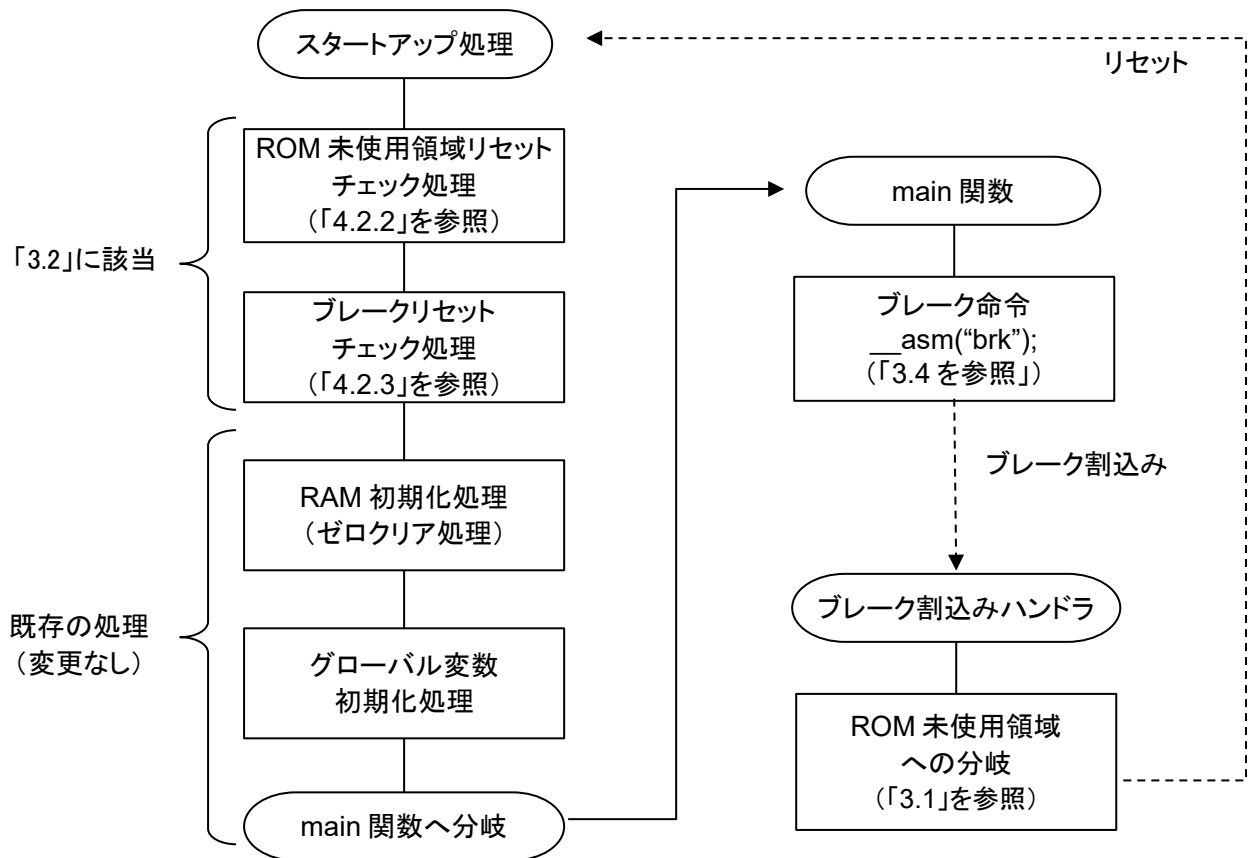
ROM 未使用領域アクセスリセットが発生したことをオンチップデバッグ未接続時でも確認できるよう、FIAR フラグが 1 にセットされていたらポートに信号を出力するなどの仕組みを入れてください。

4.2.1. プログラム例

ここでは、リファレンスボード上で動作する簡単なプログラム例を示します。
ROM 未使用領域アクセスリセットが発生したことを確認することに加え、オンチップデバッグ未接続時にブレークリセットが発生しないことも確認します。

main 関数からはブレーク命令のみを実行し、ブレーク割込みハンドラから ROM 未使用領域アクセスリセットを発生させ、それを繰り返し実行するようなフローにします。

処理フローを以下に示します。



ROM 未使用領域アクセスリセットチェック処理では、ROM 未使用領域アクセスリセットが発生していたら LED (P20) を点灯後、しばらくしたら消灯します。
ブレークリセットチェック処理では、ブレークリセットが発生していたら LED (P21) を点灯後、しばらくしたら消灯します。

上記プログラムをリファレンスボード上のマイコンにロードした後、オンチップデバッガを接続しない状態でリファレンスボードに電源を供給すると、リファレンスボード上の LED (P20) のみが点灯・消灯を繰り返すことを確認できます。

ビルド対象とするファイルは、お使いのマイコンに対応したスタートアップファイル（ASM ファイル）と main 関数が定義された C ソースファイルのみです。

スタートアップファイルでは、「3.1. スタートアップファイル中のブレーク割込みハンドラの変更」と「3.2. スタートアップファイル中のプログラム開始時処理（\$\$begin）の変更」、およびコードオプションにて WDT 動作禁止と ROM 未使用領域リセット許可を設定します。

「3.1. スタートアップファイル中のブレーク割込みハンドラの変更」については、スタートアップファイルの下記部分を変更します。

変更前	変更例
<pre> \$\$brk_reset: \$\$error: mov r0, #00h st r0, WDTMOD mov psw, #02h ;----- ; reset SFRs ; If you don't need the below then remove it. ; mov er0, #-1 st er0, BRECON0 st er0, BRECON1 st er0, BRECON2 st er0, BRECON3 ;----- bal \$; wait for Watch Dog Timer reset \$begin: </pre>	<pre> \$\$brk_reset: \$\$error: b 7:0FFFEh ; NA access reset \$begin: </pre>

「3.2. スタートアップファイル中のプログラム開始時処理（\$\$begin）の変更」については、スタートアップファイルの下記部分を変更します。なお、「3.2. スタートアップファイル中のプログラム開始時処理（\$\$begin）の変更」に該当する箇所には、ROM 未使用領域アクセスリセットが発生していたらリファレンスボード上の LED（P20）を点灯させ、ブレークリセットが発生していたらリファレンスボード上の LED（P21）を点灯させる処理を追加します。追加部分のプログラムの詳細については、「4.2.2. ROM 未使用領域リセットチェック処理」および「4.2.3. ブレークリセットチェック処理」を参照してください。

変更前	変更例
<pre> \$begin: ... ;----- ; setting Rom Window range ;----- ; nothing (fixed as range 0-0afffh) ;----- ; user SFR definition ;----- </pre>	<pre> \$begin: ... ;----- ; setting Rom Window range ;----- ; nothing (fixed as range 0-0afffh) ;----- ; checking Reset status ;----- tb FIAR bz \$\$chck_fiar_end mov r0, #0ah st r0, P2MOD0 rb P20DO mov r0, #0 mov r1, #10 \$\$chck_fiar_loop: add er0, #-1 bnz \$\$chck_fiar_loop sb FIAR sb P20DO mov r0, #0 st r0, P2MOD0 \$\$chck_fiar_end: </pre>

	<pre> tb BRKR bz \$\$chck_brkr_end mov r0, #0ah st r0, P2MOD1 rb P21DO mov r0, #0 mov r1, #10 \$\$chck_brkr_loop: add er0, #-1 bnz \$\$chck_brkr_loop sb BRKR sb P21DO mov r0, #0 st r0, P2MOD1 \$\$chck_brkr_end: \$normal: ;----- ; user SFR definition ;----- </pre>
--	--

「3.3. スタートアップファイル中のコードオプションの変更」については、スタートアップファイルの下記部分を変更します。

ROM 未使用領域アクセスリセットを許可（ビット 12 を 1（初期値））、WDT を動作禁止（ビット 0 を 0）に変更します。

<pre> ;----- ; Setting the code-option data (for ML621577) ;----- ... cseg #03 at 0ffd0h ; address dw 0fffdh ; 0ffd0h ... </pre>	<pre> ;----- ; Setting the code-option data (for ML621577) ;----- ... cseg #03 at 0ffd0h ; address dw 0fffch ; 0ffd0h, WDT disable ... </pre>
---	--

main 関数の処理には、「3.4. プログラム中のエラー処理の変更」に記載の、ブレーク命令のみを記述します。

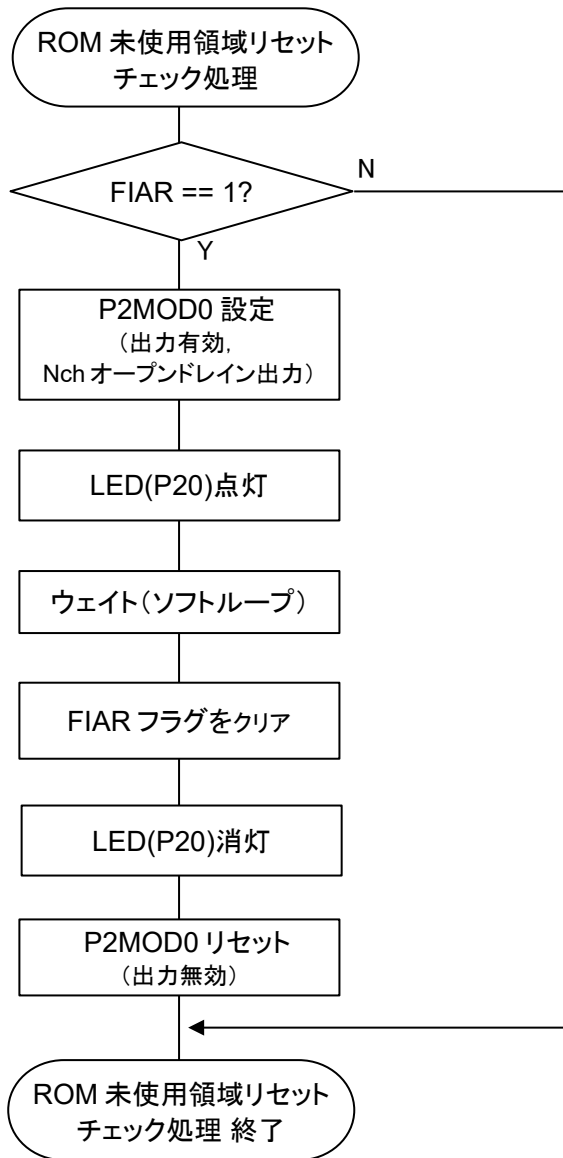
【プログラム例】

```

main.c
void main( void )
{
    __asm("brk");    /* ブレーク命令 */
}

```

4.2.2. ROM 未使用領域リセットチェック処理



プログラム例

```

; FIARフラグのチェック
tb    FIAR
bz    $$chck_fiar_end

; P2MOD0設定
mov   r0, #0ah
st    r0, P2MOD0

; LED(P20)点灯
rb    P20D0

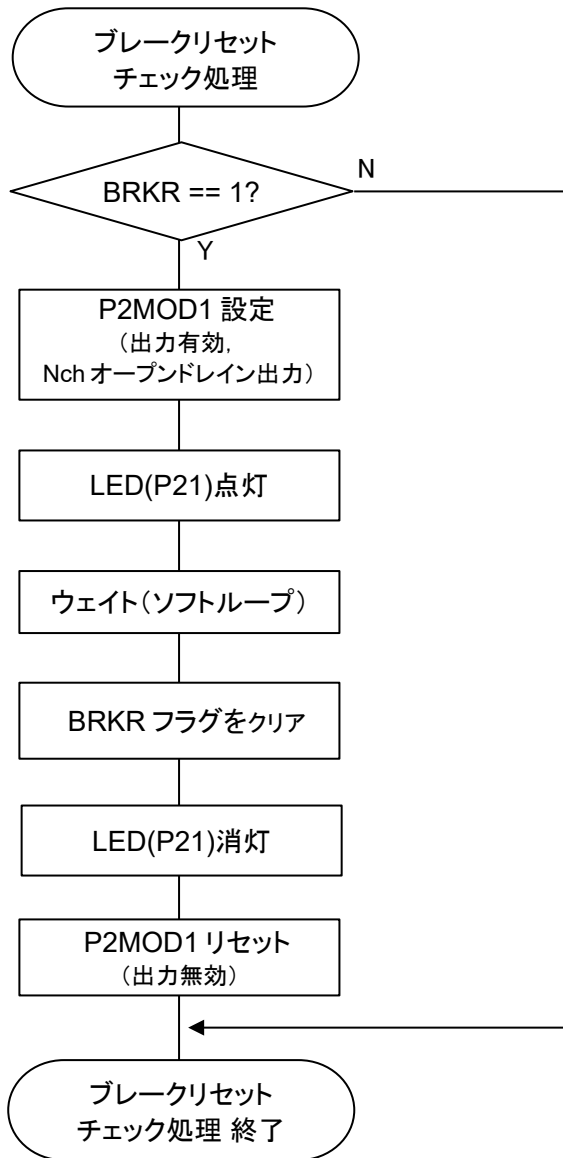
; ウェイト (ソフトループ)
mov   r0, #0
mov   r1, #10
$$chck_fiar_loop:
add   er0, #-1
bnz   $$chck_fiar_loop

; FIARフラグをクリア
sb    FIAR

; LED(P20)消灯
sb    P20D0

; P2MOD0リセット
mov   r0, #0
st    r0, P2MOD0
$$chck_fiar_end:
  
```

4.2.3. ブレークリセットチェック処理



プログラム例

```

; BRKRフラグのチェック
tb    BRKR
bz    $$chck_brkr_end

; P2MOD1設定
mov   r0, #0ah
st    r0, P2MOD1

; LED(P21)点灯
rb    P21DO

; ウェイト (ソフトループ)
mov   r0, #0
mov   r1, #10
$$chck_brkr_loop:
add   er0, #-1
bnz   $$chck_brkr_loop

; BRKRフラグをクリア
sb    BRKR

; LED(P21)消灯
sb    P21DO

; P2MOD1リセット
mov   r0, #0
st    r0, P2MOD1
$$chck_brkr_end:
  
```

4.2.4. ビルド・動作確認

【プログラムのビルド】

前項で作成したプログラム（main.c とスタートアップファイル）を LEXIDE-U16 のプロジェクトに登録してビルドします。

【プログラムのロード】

ビルドが完了したら、リファレンスボードと EASE1000 V2 をインタフェースケーブルで接続し、EASE1000 V2 と PC を USB ケーブルで接続し、リファレンスボードにプログラムをロードします。

プログラムのロードは、プロジェクトを選択した状態で、LEXIDE-U16 で DTU8 を起動（LEXIDE-U16 の[LAPIS]>[Debugger(DTU8)]）を選択します。

プログラムのロードが完了したら、DTU8 デバッガを終了します。

【プログラムの実行】

リファレンスボードから EASE1000 V2 のインタフェースケーブルを取り外し、リファレンスボードと外部電源を接続します。

リファレンスボードに電源を供給すると、リファレンスボード上の LED（P20）のみが点灯・消灯を繰り返すことを確認できます。

改版履歴

ドキュメント No.	発行日	ページ		変更内容
		改版前	改版後	
FJXT62Q1000_ NAROMRESET-01	2021.3.24	—	—	初版発行