様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

令和　３ 年　 ４ 月 ２７ 日現在

機関番号：３３９１７
研究種目：基盤研究(C)（一般）
研究期間：2017～2020
課題番号：１７Ｋ００００７５

研究課題名（和文）小規模マイコンによるIoT機器向けパーティショニングフレームワークの実現

研究課題名（英文）A partitioning framework for IoT devices using embedded microcontroller.

研究代表者
本田　晋也（Shinya, Honda）

南山大学・理工学部・教授

研究者番号：２０４０２４０６

交付決定額（研究期間全体）：（直接経費）　　3,500,000円

研究成果の概要（和文）：本研究では，リアルタイムOSと小規模なマイコンにより制御される組込みシステムを信頼性や安全性を担保しつつネットワークに接続することが可能なパーティショニングOSとセキュリティ機構を実現した。これらの機能を用いることにより，既存の組込み機器のソフトウェアの機構を大きく変えることなくネットワークに接続するIoT機器とすることが可能である。本パーティショニングOS及びセキュリティ機構は，小規模マイコン向けのハードウェアセキュリティ機構を用いることにより，実行オーバヘッドを低く押さえることに成功した。

研究成果の学術的意義や社会的意義
近年，スマートスピーカーやスマートフォンで外出先から操作可能な家電等のIoT機器の需要が高まっている。IoT機器は常にインターネットに接続されているため，インターネットからの驚異にさらされることになる。これらの驚異からシステムを守るためには，システムの重要な部分を守るパーティショニング機構や異常を検知するセキュリティ機構が必要となる。しなしながら，これらの機構は処理負荷が高く，処理能力が低いIoT機器で使用すると本来の機能が動作しなくなる。この問題を解決するためにハードウェアセキュリティ機構を用いて，処理負荷を軽減した。これにより，安心安全なIoT機器を実化することが可能となる。

研究成果の概要（英文）：In this research, we have developed a partitioning OS and security mechanism that enables an embedded system with a microcontroller to be connected to a network while ensuring reliability and safety.With these functions, it is possible to make an IoT device that connects to a network without significantly changing the software mechanism of the existing embedded system.By using a hardware security mechanism for small-scale microcontrollers, we have succeeded in reducing the execution overhead of this partitioning OS and security mechanism.

研究分野：組込みシステム

キーワード：IoT　セキュリティ　組込みシステム

様　式　C－１９，F－１９－１，Z－１９（共通）

１．研究開始当初の背景

ネットワーク技術やクラウド技術の発達と共に，組込み機器がインターネットに接続されているようになっており，これらの機器は IoT（Internet of Things）機器と呼ばれる。IoT 機器はインターネットからの脅威に晒される。IoT 機器は自動車に代表されるように人命に係わるシステムで用いられる場合があり，インターネットからの脅威に対してセキュリティを確保し高い信頼性を実現する仕組みが必要である。セキュリティを確保する方法として，パーティショニング機構（保護機構）を持つ OS を用いる方法がある。パーティショニング機構により，システム内のソフトウェアを信頼度に応じて時間的・空間的に独立したパーティションとして実行することが可能となる。その上でネットワークに係わるソフトウェア(通常系ソフトウェア)を信頼性が必要なソフトウェア(信頼系ソフトウェア)とは別のパーティションとして実行することにより，ネットワークからの攻撃により通常系ソフトウェアに不具合が発生した場合でも，信頼系ソフトウェアにその影響を及ぼすことを防ぐことが可能である。

多くの組込みシステムはリアルタイム OS と小規模なマイコンを用いて構築されている。これまで，これらの機器の多くはネットワークに接続されず，ソフトウェアも製品出荷時に固定化されるため，パーティショニング機構の必要性がなかった。 今後，小規模なマイコンを用いる機器もネットワークに接続すると予想される。これらの機器に既存のパーティショニング機構を適用するには，次の課題がある。

（1）実行オーバヘッドが大きい

組込み機器ではマイコンのメモリ保護機構をリアルタイム OS が制御することでパーティショニング機構を実現する。通常系ソフトウェアはアクセスに制約があるユーザーモードで実行し，リアルタイム OS や信頼系ソフトウェアは特権モードで動作させる。通常系ソフトウェアが OS の API 等を用いる場合には特権モードに切り替える必要があるため，実行オーバヘッドが発生する。我々の評価では，メモリ保護がない場合と比較して 3 倍程度のオーバヘッドが発生することが分かっている。

（2）既存ソフトウェアの変更量が大きい

前述の通り，通常系ソフトウェアから特権モードで動作するソフトウェアの呼び出しには，ソフトウェア割り込みによる特権モードへの変更が必要となる。パーティショニング機構がない場合はこれらの呼び出しは関数呼び出しで実現されているため，ソフトウェアの変更が必要となる。また，実行オーバヘッドの関係でリアルタイム OS によってはパーティショニング環境で通常系ソフトウェアの割込み処理をサポートしない場合がある。その場合には，割込み処理を信頼系の割込み処理と通常系のタスクに分割して実現する等の変更が必要になる。

２．研究の目的

組込みシステム向けのマイコンで大きなシェアを持つ ARM マイコンではハードウェアセキュリティ機構である TrustZone-M を備えている。TrustZone-M はパーティショニングにも使用可能であり，MPU 等の既存のハードウェアを使用した場合と比較して前述の問題を解決できる可能性が高い。本研究では，TrustZone-M を利用したパーティショニング OS の機構開発とオープンソース実装の開発を行う。具体的には次の 3 種類の項目について研究を実施する。

（1）TrustZone-M 対応のパーティショニング OS

TrustZone-M 向けのパーティショニング OS を実現する。TrustZone-M ではプロセッサに対して Secure と Non-Secure という 2 種類の論理領域が追加され，Non-Secure 側はメモリや周辺回路へのアクセスに制約を設定可能である（図 1A）。Secure 側では OS やドライバや高信頼系ソフトウェアを実行し，Non-Secure ではネットワークに関連する通常系ソフトウェアを実行する。また，Non-Secure から Secure の関数を呼び出す機構を持つため，この機構を活用して，Non-Secure 側のアプリケーションから，Secure 側の OS や信頼系ソフトウェアを呼び出す構成とすることで，実行オーバヘッド低減や既存ソフトウェアの変更を少なくする(図 1B)。

（2）セキュリティ機構

通常系ソフトウェアは保護されていないため，外部の驚異にさらされる。外部の驚異から不正な動作していないかチェックするためのセキュリティ機構を TrustZone-M により従来手法と比較して低オーバヘッドで実現する（図 1C)。

（3）時間パーティショニング

TrustZone-M はメモリやデバイスアクセスを制限する空間的なパーティションのみ実現する

ため，時間的なパーティショニング機構を過去の研究成果を活用して TrustZone-M 上に実現する(図 2)。具体的には，パーティション毎に CPU 時間と実行順序を割り当てて，システム周期毎に実行する機構を実現する。

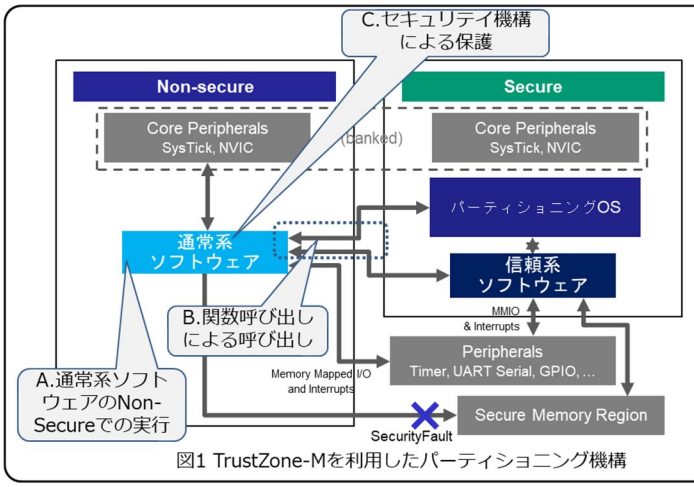


図1 TrustZone-Mを利用したパーティショニング機構

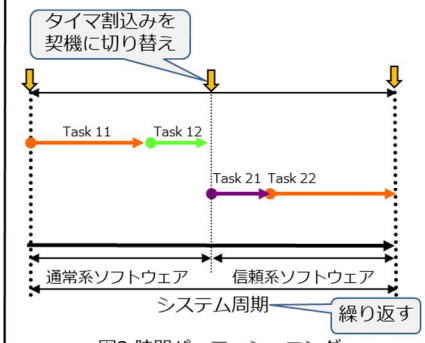


図2 時間パーティショニング

３．研究の方法

それぞれの研究テーマに関して次のように研究を実施する。

（1）TrustZone-M 対応のパーティショニング OS

既存の TrustZone-M のソフトウェアの解析を行い TrustZone-M に対するハードウェアの機構の確認や，ソフトウェアの技術方法，コンパイラの使用方法を習得する。次に，パーティショニング OS の暫定仕様を策定した後，IoT 機器の開発者にヒアリングを行うことにより，IoT 機器向けのパーティショニング OS に必要な機構を検討する。

その後，検討した機構が TrustZone-M により効率的に実現出来るか評価を行い，実現する機能を決定する。特に通常系ソフトウェアから信頼系ソフトウェアの関数の呼び出しに関しては，パーティショニングのために OS でチェックするべき事項を整理して OS で関数呼び出しをサポートする機能を用意することにより，既存のソフトウェアの変更を少なくする機構とする。

割り込みハンドラに関しても既存の通常系ソフトウェアの割込みハンドラを変更することなくパーティショイング環境で実行するための機構を検討する。具体的には TrustZone-M が持つ Non-Secure で割込みハンドラを実行する機構を用いるのではなく，割込みを一旦 Secure で受け付け，時間監視用のタイマをスタートさせてから，関数呼び出しで Non-Secure 側の割込みハンドラ関数を呼び出す。割込みハンドラは終了すると関数リターン処理を行い Secure 側に戻る。設定された時間以内で割込みハンドラがリターンしなかった場合には，Secure 側にタイマ割込みが入り，Non-Secure の処理を打ち切りエラーとする機構とする。

次に検討した機構の実装を行う。実装は既存の ITRON 仕様のリアルタイム OS である TOPPERS/ASP カーネル(ASP カーネル)をベースとする。ITRON 仕様は日本で広く使用されているリアルタイム OS 仕様である。

実装後は，リアルタイム OS の API 実行時間や信頼系の関数呼び出し実行オーバヘッド等を計測する。評価の比較対象としては，既存のメモリ保護ハードウェアである MPU を用いた ITRON 仕様のリアルタイム OS である TOPPERS/HRP2 カーネルと比較を実施する。API の実行時間に関しては，非特権タスクから API を呼び出し場合の実行時間をそれぞれ計測して評価する。関数呼び出し実行オーバヘッドに関しては，HRP2 カーネルでは信頼関数として登録して呼び出し実行オーバヘッドを計測する。提案するパーティショニング OS では実現した機構により呼び出すことで，両者を計測して評価を行う。

（2）セキュリティ機構

セキュリティ機構としては，軽量なコントロールフローインテグリティ（CFI）の実現技術を実現する。CFI は，コントロールフロー攻撃に対してセキュリティを確保するための技術の１つであるが，従来の CFI の実装は，マイコンを用いた組込みシステムには適用できなかった。本研究では，LLVM コンパイラの拡張と，Armv8-M の TrustZone 機能を用いてシャドウ例外スタックを実現するモニタにより，軽量な CFI を実現する手法を実現する。また，提案した手法を実装し，性能評価により，従来手法よりも小さいオーバヘッドで実現できることを評価する。

（3）時間パーティショニング

過去の研究を元に TrustZone-M 向けの機構を検討する。時間パーティショニングはパーティション毎に時間を割り付けて実行する。提案者はこれまで MPU を用いたメモリ保護に組み合わせたリアルタイム OS の開発の経験があるため，これを基本に TrustZone-M を利用したパーティショニング OS に適用する。次に検討した時間パーティショニングの機構を実装し，事例を用

いて評価を行う。事例は実際の IoT 機器を想定して，ネットワークからの攻撃への耐性や実行オーバヘッドについて評価する。

４．研究成果

それぞれの研究テーマの研究成果は次の通りである。

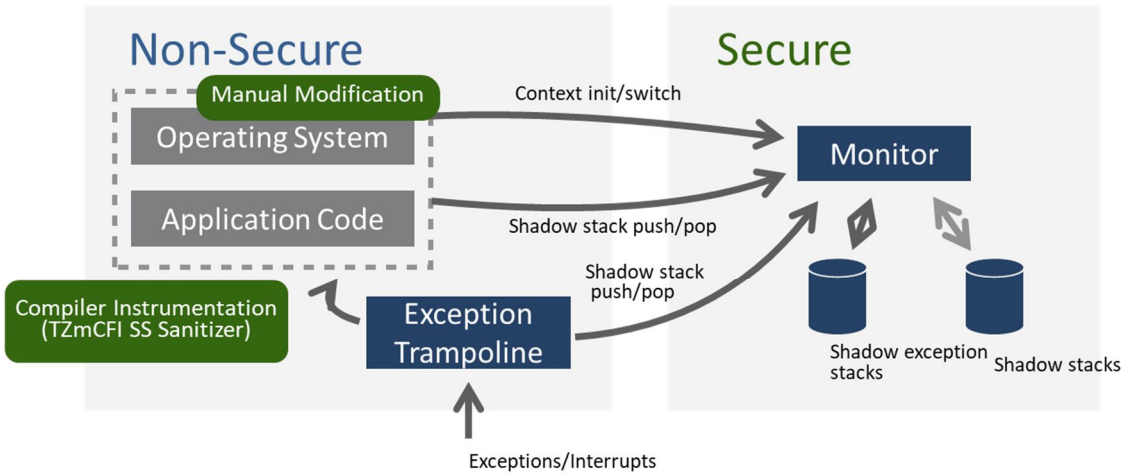
（1）TrustZone-M 対応のパーティショニング OS

TrustZone-M ハードウェア及び，既存の TrustZone-M のソフトウェアの解析を行い TrustZone-M に対するハードウェアの機構の確認や，ソフトウェアの技術方法，コンパイラの使用方法を習得した。次に，パーティショニング OS の暫定仕様を策定して，実現する機能を決定した。そして，それらの機能を低オーバヘッドで実現可能な実現方法を検討した。特に通常系ソフトウェアから信頼系ソフトウェアの関数の呼び出しに関しては，パーティショニングのために OS でチェックするべき事項を整理して OS で関数呼び出しをサポートする機能を用意することにより，実行オーバヘッドと既存のソフトウェアの変更を少なくする機構を実現した。

実現した機構以外の実現方法として，セキュアライブラリ方式，デュアル OS 方式を検討し，それらと提案手法の定性的な比較評価を実施し，提案手法はメモリ・実行オーバヘッド及び開発効率の観点でメリットがあることを示した。また，既存の RTOS やメモリ保護 RTOS との定量的な評価を実施した。評価の結果，OS-API の実行時間や割込み応答時間に関しては，提案手法は通常の RTOS に近い性能であることを示した。また，提案手法の通常の RTOS からのコード変更量は，2 割程度となり，特にプロセッサに依存しない部分の変更は 30 行程度と非常に小さいことを示した。提案機構と評価結果をまとめて論文として発表した。

（2）セキュリティ機構

CFI と呼ばれるプログラムの 関数呼び出しとリターンが設計通り実行されているかチェックする機構を TrustZone-M ハードウェアを用いて低オーバヘッドでの実現方法を検討し，実装して評価を行った。具体的には，シャドースタックと呼ばれる，関数の呼び出し履歴の保存領域を TrustZone-M を用いた Secure 領域に保存するランタイムを用意し，アセンブラを変更してアセンブリ時に関数呼び出しの箇所に Secure 側のランタイムの呼び出しコードを挿入する。これにより，関数呼び出しをシャドースタックに保存し，関数からのリターン時にも Secure 側が呼び出されることにより，関数呼び出しとリターンの正当性をチェックする。また，この機構を割込みの入口処理にも適用する手法を検討し実現した。評価の結果，低実行オーバヘッドで実現できていることを確認した。これらの成果は国内会議及び国際会議で発表し，論文誌に論文として発表した。



（3）時間パーティショニング

マルチコアで時間パーティショニングを実現する機構を検討し，RTOS に実装した（時間パーティショニング OS を）。実装した RTOS はオープンソースとして公開した。また，実現した時間パーティショニング機構を評価するための性能評価スイートを開発して，その評価結果を国内学会で発表した。

５．主な発表論文等

〔雑誌論文〕　計2件（うち査読付論文　2件／うち国際共著　0件／うちオープンアクセス　1件）

１．著者名 河田智明, 本田晋也	４．巻 59
２．論文標題 ARM TrustZone for ARMv8-Mを利用した軽量メモリ保 護RTOS	５．発行年 2018年
３．雑誌名 情報処理学会論文誌	６．最初と最後の頁 762-774
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 －

１．著者名 T. Kawada, S. Honda, Y. Matsubara, H. Takada	４．巻 -
２．論文標題 TZmCFI: RTOS-Aware Control-Flow Integrity Using TrustZone for Armv8-M	５．発行年 2020年
３．雑誌名 International Journal of Parallel Programming, Springer	６．最初と最後の頁 -
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 －

〔学会発表〕　計5件（うち招待講演　0件／うち国際学会　1件）

１．発表者名 T. Kawada, S. Honda, Y. Matsubara, H. Takada
２．発表標題 Shadow Exception Stacks: Control-Flow Integrity For Asynchronous Exceptions Using TrustZone For Armv8-M
３．学会等名 The 6th International Embedded Systems Symposium(IESS2019)（国際学会）
４．発表年 2019年

１．発表者名 本田晋也 山本椋太
２．発表標題 車載制御システム向け次世代プロセッサの 仮想化支援機能を用いたハイパーバイザー
３．学会等名 情報処理学会研究報告
４．発表年 2019年

１．発表者名
河田智明，本田晋也，松原豊，高田広章

２．発表標題
Arm TrustZone for Armv8-Mを利用したマルチタスク対応CFIの検討

３．学会等名
情報処理学会 組込みシステムシンポジウム

４．発表年
2018年

１．発表者名
河田智明，本田晋也，松原豊，高田広章

２．発表標題
Shadow exception stacks: 非同期例外を対象としたTrustZone-MベースのCFI機構

３．学会等名
暗号と情報セキュリティシンポジウム

４．発表年
2019年

１．発表者名
手塚湧太郎,本田晋也,大谷寿賀子,枝廣正人

２．発表標題
時間パーティショニング機構を持つリアルタイムOSの性能評価手法

３．学会等名
第55回組込みシステム研究発表会,情報処理学会

４．発表年
2020年

〔図書〕　計0件

〔産業財産権〕

〔その他〕

―

６．研究組織

	氏名 （ローマ字氏名） （研究者番号）	所属研究機関・部局・職 （機関番号）	備考

７．科研費を使用して開催した国際研究集会

〔国際研究集会〕　計0件

８．本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関