Safety und Security im Code

AUTOR

Dr. Darren Buttle ist Produktmanager ASCET bei der **ETAS GmbH**.

ESDL als Basis für sichere Software

Für Embedded Software ist die Programmiersprache C das Maß der Dinge. Aber die Erstellung von sicherem C-Code ist nicht einfach. Doch je autonomer die Fahrzeuge werden, umso größer werden die Anforderungen an die Integrität der Software. Darum hat ETAS die Embedded Software Development Language (ESDL) entwickelt. Damit schaffen es Entwickler, mehr Software schneller zu entwickeln, die unter anderem die Standards ISO 26262 und IEC 61508 erfüllt.

In den letzten 40 Jahren hat sich C zur De-facto-Programmiersprache für die Entwicklung eingebetteter Software entwickelt. C ist einfach, kompakt, schnell, portabel und wird von vielen Tools unterstützt.

Doch C hat auch Tücken. Allzu leicht schleichen sich Fehler ein, die im Code nur schwer zu finden sind. Die Syntax macht es leicht, Fehler beim Programmieren zu machen – optionale Klammern, Zuweisungen in Ausdrücken und automatisches „switch/case-Durchfallen“ sind nur einige Beispiele. Zusätzlich gibt es semantisch zweifelhafte oder komplexe Features, die nur schwer richtig zu verwenden sind und dazu verleiten, „an der Grenze zur Sicherheit“ zu programmieren, beispielsweise Sprünge, Zeiger und integrale Promotion. Auch diese Aspekte können gefährlich zusammenwirken.

Durch die Verwendung von Programmierrichtlinien wie MISRA-C und CERT-C können viele dieser Risiken vermieden werden. Aber selbst bei Einhaltung dieser Richtlinien können weiterhin Probleme auftreten. Richtlinien verhindern weder Laufzeitprobleme wie Speicherüberlauf noch numerische Probleme wie Überlauf oder Division durch Null. Richtlinien können auch keine unsinnigen Operationen erkennen, wenn beispielsweise eine Geschwindigkeit einen Grenzwert überschreitet, eine Temperatur unter den absoluten Tiefpunkt fällt oder versehentlich eine Entfernung zu einem Druck addiert wird. C ist nicht expressiv genug, um diese Informationen zu erfassen und kann diese folglich auch nicht überprüfen.

Um diese Probleme zu vermeiden, sind zusätzliche Maßnahmen erforderlich, zum Beispiel statische Analysen und Testläufe zur Fehlerbehebung. Das ist ineffizient. Viel besser wäre es, Fehler von Anfang an zu vermeiden, damit sie gar nicht erst beseitigt werden müssen.

Eine bessere Programmiersprache

Die ETAS-Antwort auf all diese Herausforderungen lautet Embedded Software Development Language (ESDL) – eine neue Programmiersprache zur effektiven Erstellung sicherer Software. ESDL vermeidet die typischen Fallstricke der C-Programmierung, erleichtert die Wiederverwendung von Software, vereinfacht die Wartung und unterstützt Variantenverwaltung. Mit ESDL können Entwickler sich wieder mehr auf ihre Kernaufgaben konzentrieren und müssen nicht mehr um die Unzulänglichkeiten von C herum programmieren.

C-Code mithilfe von Codegenerierung

Die effiziente Verwendung von ESDL wird durch ETAS ASCET-DEVELOPER 7 (siehe Seite 15), einer Eclipse-basierten integrierten Entwicklungsumgebung (IDE) und einem C-Codegenerator, ermöglicht.

Die IDE bietet moderne Bearbeitungsfeatures wie Sprachtemplates, Codevorschläge und Quick Fixes für Probleme. Dadurch ist ESDL auch für Anfänger leicht zu lernen. ASCET-DEVELOPER 7 prüft kontinuierlich auf ESDL-Programmierfehler, berechnet Qualitätsmetriken und weist auf Best Practices hin. Entwickler erhalten Feedback „on-the-fly“ bereits bei der Bearbeitung. Zwischen der Eingabe eines Codierfehlers und seinem Auffinden vergeht praktisch keine Zeit.

Der C-Codegenerator übersetzt ESDL in MISRA-konformen C-Code. ASCET-DEVELOPER 7 fügt automatisch defensive Codeüberprüfungen an den Stellen ein, an denen sie für die Laufzeitsicherheit erforderlich sind. Damit müssen diese nicht von Hand ergänzt und verwaltet werden. Der generierte C-Code lässt sich leicht in einen bestehenden C-basierten Entwicklungsprozess integrieren.

Programmiersprache gegen mögliche Fehler absichern

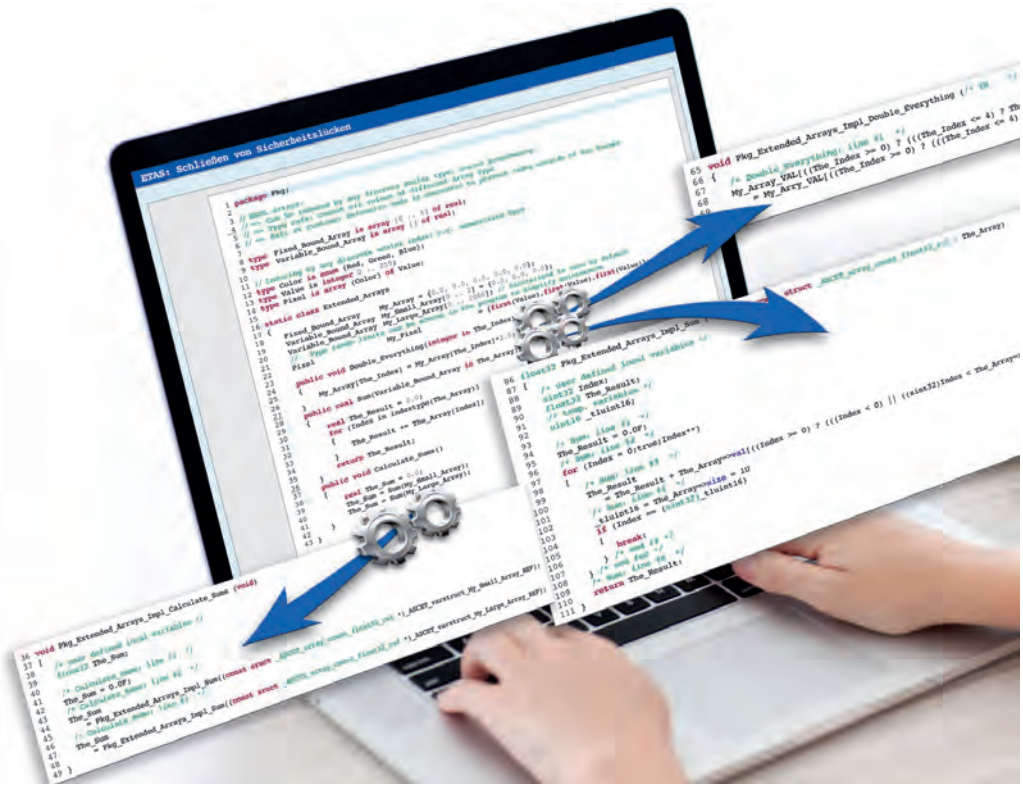
ESDL deckt viele Aspekte ab, die bereits in den C-Programmierrichtlinien enthalten sind. Zudem enthält das Design von ESDL Features, die es einfach machen, die Sprachanforderungen in Standards wie ISO 26262 und IEC 61508 zu erfüllen. Durch diese Konzepte in ESDL kann ASCET-DEVELOPER 7 mehr Fehlerfälle während der Bearbeitung überprüfen, als es in der klassischen C-Programmierung möglich ist.

Die ESDL-Syntax ist der von C sehr ähnlich, weshalb die Entwickler sich damit sofort vertraut fühlen. ESDL vermeidet jedoch die gefährlichen C-Features, die Richtlinien normalerweise einschränken oder verbieten. In ESDL gibt es keine optionalen Klammern, keine Anweisungen in Ausdrücken, keine Zuweisung an Schleifenvariablen, kein automatisches switch/case-Durchfallen, keine implizite integrale Promotion, keine globalen Variablen, keine Zeiger, keine Sprungfunktion, keine Unions usw. Das Fehlen all dieser Fallstricke macht ESDL schon in sich sicherer als C.

Alle ESDL-Berechnungen sind frei von numerischen Problemen wie Overflow, Underflow, Division durch Null oder Signed Overflow.

Array-Zugriffe außerhalb ihres Bereichs sind in ESDL nicht möglich: Das typische Pufferüberlaufproblem, bekannt aus vielen Sicherheitsberichten, kann mit ESDL nicht auftreten.

ESDL verfügt über ein umfangreiches Typsystem, welches einem Typen einen Namen zuweist (ähnlich wie



Sicherheitslücken können sofort beim Programmieren geschlossen werden.

ein „typedef" in C), aber mit weiteren Informationen zum zulässigen Wertebereich und (optional) zur erforderlichen Auflösung. In ESDL kann beispielsweise eine Geschwindigkeit als eine Fließkommazahl im Bereich von 0,0 bis 260,0 km/h mit einer Auflösung von 0,01 km/h definiert werden. Typen können Einheiten, zum Beispiel Meter, Grad oder Zeit, verwenden.

Die Kompatibilität der Einheiten wird automatisch überprüft und verhindert, dass beispielsweise eine Zeit zu einer Entfernung addiert wird.

Der Codegenerator von ASCET-DEVELOPER 7 verwendet die ESDL-Typeninformation, um den speicheroptimalen C-Typ auszuwählen und um defensive Code-Überprüfungen zur Laufzeit zu erstellen, welche die Plausibilität der Werte jederzeit sicherstellen.

Änderungen können leicht an einer Stelle im Programm vorgenommen werden und systematisch durch erneute C-Codegenerierung angewendet werden. Ein weiterer Vorteil: Reviews und Inspektionen sind mit ESDL einfacher. ESDL-Programme sind nicht überfrachtet mit händischen Bereichsüberprüfungen, die ein Programm unübersichtlich machen und ein einfaches Verstehen der Programmabläufe erschweren.

Datenzugriff kontrollieren und Wiederverwendung ermöglichen

ESDL ist objektbasiert und verwendet Klassen, um den Datenzugriff zu verwalten und zu kontrollieren. Objekte können sicher verwendet werden und verfügen über definierte Speicherbegrenzungen. Im Gegensatz zu C++ und Java gibt es bei ESDL keine Speicherlecks, weil auf eine dynamische Speicherzuweisung verzichtet wird.

Die ESDL-Klassen unterstützen auch Produktlinienvarianten, ohne dass „Clone & Own" von Funktionsverhalten erforderlich wäre. Varianten sind möglich für:

- Code
- Dateninitialisierung
- Speicherzuweisung
- C-Speicherdarstellung (zum Beispiel Wechsel zwischen Gleit- und Festkommadarstellung)

Die Datenkonsistenz in einer Echtzeitumgebung wird in ESDL über einen Thread-sicheren Kommunikationsmechanismus über „Messages" realisiert. Für diese werden Lese- und Schreibzugriffe klar definiert und damit ein unzulässiger Zugriff auf Daten verhindert.

Fazit

Im komplexen Entwicklungsumfeld zunehmend vernetzter Fahrzeuge kann die Flexibilität der Programmiersprache C zum Nachteil werden. Zu leicht schleichen sich unentdeckt Fehler in den Code ein und es ist zu zeitaufwendig und zu ineffizient, diese Fehler später im Entwicklungsprozess zu beheben. Bei der Arbeit mit C wird viel Zeit damit verbracht, die Fallstricke von C zu umgehen.

Mit ESDL und ASCET-DEVELOPER 7 ermöglicht ETAS es, sicheren C-Code effektiver und effizienter zu erstellen. ESDL beseitigt ganze Kategorien möglicher Fehlerquellen und erleichtert die Wiederverwendung von Software und generiertem Code über mehrere Projekte hinweg. Dank ESDL erreicht die Entwicklung von Embedded Software ein neues Niveau an Effizienz und Sicherheit.