



AEROSPACE INFORMATION REPORT

AIR6027™

REV. A

Issued 2012-05
Stabilized 2017-06

Superseding AIR6027

Considerations for Safe Store Operation on Manned and Unmanned Vehicles

RATIONALE

This document has been determined to contain basic and stable technology which is not dynamic in nature.

STABILIZED NOTICE

This document has been declared "Stabilized" by the SAE AS-1B Aircraft Store Integration Committee and will no longer be subjected to periodic reviews for currency. Users are responsible for verifying references and continued suitability of technical requirements. Newer technology may exist.

SAE Technical Standards Board Rules provide that: "This report is published by SAE to advance the state of technical and engineering sciences. The use of this report is entirely voluntary, and its applicability and suitability for any particular use, including any patent infringement arising therefrom, is the sole responsibility of the user."

SAE reviews each technical report at least every five years at which time it may be revised, reaffirmed, stabilized, or cancelled. SAE invites your written comments and suggestions.

Copyright © 2017 SAE International

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of SAE.

TO PLACE A DOCUMENT ORDER:
Tel: 877-606-7323 (inside USA and Canada)
Tel: +1 724-776-4970 (outside USA)
Fax: 724-776-0790
Email: CustomerService@sae.org
<http://www.sae.org>

SAE values your input. To provide feedback on this Technical Report, please visit <http://standards.sae.org/AIR6027A>

1. SCOPE

The information presented in this AIR is intended to provide designers of armed unmanned systems with guidelines that may be applied to ensure safe integration and operation of weapons on unmanned platforms. The guidelines have been developed from experiences gained in the design and operation of weapons on manned aircraft that have been accepted by relevant safety authorities in the USA and Europe and proven effective over many years. Whilst the guidelines have been developed from experience with aircraft operations, the concepts are considered equally applicable to non-aircraft systems, such as those used on the surface or undersea environments.

This document does not attempt to define or describe a comprehensive safety program for unmanned systems.

System Safety is a system characteristic and a non-functional requirement. It has to be addressed at each level of system design, system integration and during each phase of system operation. System safety is achieved when the system operation does not cause inadvertent personnel injuries, destruction of the system or damage to the environment.

Section 3 of the document contains an introduction to methods by which the safety of a system can be assessed.

Section 4 describes the safety principles commonly applied to the design and operation of weapons on manned aircraft.

Section 5 describes how the safety principles established for manned aircraft may be applied to unmanned systems,

Section 6 provides conclusions and recommendations.

2. APPLICABLE DOCUMENTS

The following publications form a part of this document to the extent specified herein. The latest issue of SAE publications shall apply. The applicable issue of other publications shall be the issue in effect on the date of the purchase order. In the event of conflict between the text of this document and references cited herein, the text of this document takes precedence. Nothing in this document, however, supersedes applicable laws and regulations unless a specific exemption has been obtained.

2.1 SAE Publications

Available from SAE International, 400 Commonwealth Drive, Warrendale, PA 15096-0001, Tel: 877-606-7323 (inside USA and Canada) or 724-776-4970 (outside USA), www.sae.org.

AS5725	Interface Standard, Miniature Mission Store Interface
AS5726	Interface Standard, Interface for Micro Munitions
ARP4754	Guidelines for Development of Civil Aircraft and Systems
ARP4761	Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment

2.2 IEEE Publications

Available from Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854-1331, Tel: 732-981-0060, www.ieee.org.

IEEE STD 1228-1994	Standard for Software Safety Plans
--------------------	------------------------------------

2.3 RTCA Publications

Available from Radio Technical Commission for Aeronautics Inc., 1828 L Street, NW, Suite 805, Washington, DC 20036, Tel: 202-833-9339, www.rtca.org.

RTCA DO-178	Software Considerations in Airborne Systems and Equipment Certification
-------------	---

2.4 U.S. Government Publications

Available from the Document Automation and Production Service (DAPS), Building 4/D, 700 Robbins Avenue, Philadelphia, PA 19111-5094, Tel: 215-697-6257, <http://assist.daps.dla.mil/quicksearch/>.

MIL-STD-882	Standard Practice For System Safety
MIL-STD-1553	Interface Standard For Digital Time Division Command/Response Multiplex Data Bus
MIL-STD-1629	procedures for performing a failure mode, effects and criticality analysis (cancelled)
MIL-STD-1760	Aircraft/Store Electrical Interconnection System
MIL-HDBK 516	Airworthiness Certification Criteria

2.5 Other Publications

N. G. Levinson A New Approach To System Safety Engineering

Various Authors System Software Safety Handbook (2010)

3. RISK ASSESSMENT, HAZARD ANALYSIS AND SOFTWARE SAFETY CLASSIFICATION

This section provides an overview of the analyses that must be carried out when designing the safety characteristics of a system. It should be noted that there are many documents containing guidelines and requirements applicable to system safety design for the operation of weapons on platforms. Adherence to any one document will not necessarily ensure compliance with others. Documents applicable to the operation of stores on platforms include: MIL-STD-882, IEEE STD 1228, SAE ARP4761 and SAE ARP4754.

In general, it is important that when designing the safety characteristics of any system, reference should be made to the complete set of safety documents relevant to the system. A common understanding of the set of required safety characteristic should be reached with the safety authorities. It should also be noted that there is likely to be more than one safety authority associated with the operation of stores on a platform. There may be separate authorities covering explosives and fuzing, store safety, platform safety, airworthiness and clearances for flying in controlled airspace in the case of air platforms. Each nation or armed service may have its own organization holding these authorities.

In addition, caution should be exercised in the use of terminology due to the lack of commonality across the documents. As an example, MIL-STD-882 Rev D contains very specific definitions of the words 'hazard' and mishap' that are not necessarily contained in other relevant documents. As an example the banana peel is the hazard, slipping on it is the mishap. In this AIR, the term 'failure' is used to encompass all instances of the system not operating as expected, including mishaps or the arising of hazards that were (or were not) previously identified, as defined in MIL-STD-882.

The following paragraphs describe some of the approaches and processes commonly used when designing and assessing the safety characteristics of store operation on manned aircraft. They are offered for consideration when designing unmanned systems.

3.1 Risk Assessment Procedure

The system safety assessment is a continuous systems engineering process applied during the whole system life cycle. Hazard Analyses and Safety Assessments using different techniques, including software safety analyses, are performed on all system development levels to the level of rigor required to ensure robustness and correct operation of the system.

The first step in mitigating safety risks is early identification of failure modes to which the design of the aircraft and store system can contribute.

- Causes of potential failures are identified by a combination of activities: analytical methods, including, but not limited to Functional Hazard Assessments, Fault Tree Analysis, Failure Modes, Effects and Criticality Analysis,
- experience from legacy programmes, and
- identification of new failures and their causes during the development of the system.

Risk assessment is performed to establish safety risks to the weapon system caused by functional failure of system component(s) (which includes operator actions) or faults and failure conditions. Functional Hazard Assessments and Preliminary Hazard Analyses, and System safety assessments, using blended approaches and methods from SAE ARP4761, SAE ARP4754, MIL-STD-882, and IEEE STD 1228, when combined with effective system safety techniques should identify failure modes and enable risk mitigation. This is often through derived requirements that specify safety features in the design.

Safety Critical (SC) Functions associated with weapons ready, weapons solutions and/or release and control that are commanded, controlled and monitored by software will require a software safety effort using IEEE STD 1228-1994 or equivalent alternative methods.

Each (system) function must be evaluated with respect to the effect of the function failure on the mission, platform/personnel and environment. The principal failure modes are:

- Loss of function
- Provision of function when not required
- Provision of function incorrectly
- Hazardously misleading information that could lead to malfunction or human error

NOTE: MIL-STD-1629A (cancelled) provides more detailed failure modes:

- Premature operation
- Failure to operate at prescribed time
- Intermittent operation
- Failure to cease operation at prescribed time
- Loss of output of failure during operation
- Degraded output or operational capability
- Other unique failure conditions, as applicable, based upon system characteristics and operational requirements or constraints.

System boundary conditions are established to focus on the relevant activities within the system being analyzed. These activities may include operator actions, and some initial conditions, which best describe the system in a fault-free state. Initial conditions are therefore steady state events, which are normally expected and directly related to the circumstances for which the analysis (e.g., fault tree) is performed.

The effect of the failure on other subsystems must be considered, contributing factors (e.g., maintenance, operational, or environmental influences, etc.) including secondary failures, which may influence the mishap severity must be identified.

The determination of a mishap risk requires consideration of two aspects: the probability of occurrence and the consequences.

Consideration of the consequences of a mishap occurring usually assumes a number of levels of severity. The following levels are extracted from MIL-STD-882:

- Catastrophic: causing death and/or platform loss
- Critical: causing severe injury and/or major platform damage
- Marginal: causing minor injury and/or minor platform damage
- Negligible: causing less than minor injury and/or platform damage

Other standards may expand these definitions; for example, by taking into account the damage to the environment.

Table 1 provides an example of the approval authority required for acceptance of the combination of mishap probabilities and consequences:

TABLE 1 - MISHAP RISK ACCEPTANCE AUTHORITY

	Severity			
	Catastrophic	Critical	Marginal	Negligible
Probability				
Frequent ($\geq 10^{-1}$)				
Probable ($<10^{-1}$ to $\geq 10^{-2}$)				
Occasional ($<10^{-2}$ to $\geq 10^{-3}$)				
Remote ($<10^{-3}$ to $\geq 10^{-6}$)				
Improbable ($<10^{-6}$)				

	HIGH LEVEL ACCEPTANCE AUTHORITY
	MEDIUM LEVEL ACCEPTANCE AUTHORITY
	LOW LEVEL ACCEPTANCE AUTHORITY

3.2 Software Safety, Software Functions and Classification

Whilst hardware failures usually have a probability of occurrence, system failures caused by software do not: there either is or is not a software error. The integrity of software is controlled by the software development process, i.e., dependent on the criticality (or failure condition category) of a function realized in software, a software development class is determined, which specifies the application of the respective software development level (see also for example, RTCA DO-178). The mishap severity category "Catastrophic" will always require the highest software integrity level, regardless of the standard. For example DO-178B level A or DEF-STAN -00-56 SIL/SIR 4 is required for any software function that could contribute to a catastrophic failure condition/hazard. Levinson states that most of the system failures caused by software can be derived from failures in the respective requirement(s).

While following DO-178B provides some confidence in software assurance, development, and verification rigor it does little to ensure safety requirements and critical functions implemented in software are implemented correctly and function correctly to prevent, eliminate and control hazards. Safety and airworthiness qualifications and certification shall have objective evidence that software with complexity and criticality have been analyzed and verified from a safety perspective. The safety aspects of software and functions allocated under control of safety-critical software shall follow the industry standard for software safety as promulgated in IEEE STD 1228-1994. These include the following six standard tasks widely used.

1. Software Safety Requirements Analyses (establish safety functions implemented in software)
2. Software Safety Design Analysis (top level and architectural design)
3. Software Safety Design Analysis (detailed functional design)
4. Software Safety Code Analysis (selective at unit level, not pervasive)
5. Software Safety Test Analysis (to ensure test cases for safety verification)
6. Software Safety Change Analysis (to assess safety impacts of software changes)

It should be noted that for airworthiness (MIL-HDBK-516B) and safety aspects of software purposes, following IEEE STD 1228 will help yield objective safety evidence that can be used to meet risk, requirements and airworthiness/certification criteria. Additionally, the software safety analyses along with other detailed system safety documentation can be used in data package submittal to help satisfy the UK MOD and US DoD Ordnance Boards or equivalent. The US Navy Weapon System Explosive Safety Review Board (WSESRB) requires software safety work products be provided to the Software System Technical Review Panel (SSSTRP). The US Air Force Nonnuclear Munitions Safety Board (NNMSB), US Army Fuze Board and MoD Defence Ordnance Safety Group all require substantiating and objective evidence of acceptable safety risk on any weapons and often on the release and control aspects of those weapons. Therefore, if software has a significant impact on system functions, an appropriate software safety effort will be required.

For further considerations, it is helpful to distinguish between safety critical functions – in case of a failure, the function directly causes a catastrophic or critical mishap – and safety involved functions – a function, where failure can cause catastrophic, critical or marginal mishaps only in combination with other independent failure(s).

The determination of the classification required to develop safety critical software (e.g., per DO-178) could be enhanced by the introduction of functions or functional contributions realized in hardware: either a single or double software error causes a mishap or the combination of a software error with an independent event (hardware failure or wrong crew action) leads to the mishap. The latter combines e.g., hardware failure rates – the independent event – with software failures causing a specific mishap (i.e., a software error together with a hardware failure causes the loss of the aircraft). The combination of a hardware failure and software error is then transferred into a software safety classification level (see the System Software Safety Handbook for further reference). Where a function causes a specific hazard only in combination with another independent function, the safety criticality can be reduced.

As a core part of the safety assessment, the independence argument for software functions or software functions implemented together with hardware must be prepared.

Functional independence must be addressed and demonstrated for the following cases (Table 2):

TABLE 2 – FAILURE AND SECONDARY FAILURE POSSIBILITIES

Failure Source	Secondary Failure
Pilot or event	Hardware
Pilot or event	Software
Hardware	Software
Software	Hardware
Software	Software

- a. Pilot/event and hardware are independent, because neither the hardware nor the pilot is affecting each other. Also environmental/operational events do not affect each other. This is e.g., true for a system, which is ergonomically designed and does not put additional workload on the pilot due to bad design.
- b. Pilot/event and software are independent when software failures identified during the requirements hazard analysis does not reduce pilot abilities to perform actions. However, the ergonomics/procedural requirements are also valid for this case.
- c. Hardware and hardware are independent, when a failure in hardware 1 does not affect hardware 2 (exclusion of common mode failures).
- d. Software and Hardware are independent when a software failure does not cause a hardware failure and vice versa.
- e. Software and Software independence is more difficult to determine. The demonstration of independence will be done firstly by establishing the software development process required by the software class, which has been determined based on the severity of a hazard a software failure could cause. Software development may consider the following processes to establish the independence argument:

1. Diversity

Independence of two software functions is determined top down, starting from the definition of fundamentally diverse functions, realized in software so that a common fault cannot cause both software functions to fail and therefore cause the hazard.

2. Top down hazard analysis

This must be done in order to prevent the diversity of functions by the software. This can be avoided if the hazard analysis is done starting from the requirements hazard analysis down to code level hazard analysis. The following questions arise.

- are any signals/data common – does the corruption of a common signal cause a hazardous failure of both functions?
- are software components common – does any combination of software failures cause a hazardous failure?
- does the software share the same physical environment? For instance in cases where both software functions share the same processor and memory, a failure of the hardware would cause both software functions to fail. Thus the sharing of processor and memory is not an option for safety critical functions, two processor boards with their own memory have to be introduced. However, if the common power supply fails, the issue is still not resolved.