

ANSI/ISA-TR88.00.02-2015

A Technical Report prepared by ISA and registered with ANSI

Machine and Unit States: An implementation example of ANSI/ISA-88.00.01

Approved 26 October 2015

ANSI/ISA-TR88.00.02-2015, Machine and Unit States: An implementation example of
ANSI/ISA-88.00.01

ISBN: 978-1-941546-65-9

Copyright © 2015 by ISA. All rights reserved. Not for resale. Printed in the United States of
America.

ISA
67 Alexander Drive
P. O. Box 12277
Research Triangle Park, NC 27709 USA

Preface

This preface, as well as all footnotes, is included for information purposes only and is not part of ANSI/ISA-TR88.00.02-2015.

This technical report has been prepared as part of the service of ISA, the International Society of Automation, toward a goal of uniformity in the field of instrumentation. To be of real value, this document should not be static but should be subject to periodic review. Toward this end, the Society welcomes all comments and criticisms and asks that they be addressed to the Secretary, Standards and Practices Board; ISA, 67 Alexander Drive; P.O. Box 12277; Research Triangle Park, NC 277099; Telephone (919) 549-8411; Fax (919) 549-8288; E-mail: standards@isa.org. This ISA Standards and Practices Department is aware of the growing need for attention to the metric system of units in general, and the International System of Units (SI) in particular, in the preparation of instrumentation standards, recommended practices, and technical reports. The Department is further aware of the benefits of USA users of ISA standards of incorporating suitable references to the SI (and the metric system) in their business and professional dealings with other countries. Toward this end, the Department will endeavor to introduce SI and acceptable metric units in all new and revised standards to the greatest extent possible. The Metric Practice Guide, which has been published by the Institute of Electrical and Electronics Engineers (IEEE) as ANSI/IEEE Std. 268-1992, and future revisions, will be the reference guide for definitions, symbols, abbreviations, and conversion factors.

It is the policy of ISA to encourage and welcome the participation of all concerned individuals and interests in the development of ISA standards. Participation in the ISA standards-making process by an individual in no way constitutes endorsement by the employer of that individual, of ISA, or of any of the standards, recommended practices, and technical reports that ISA develops.

This standard is structured to follow the IEC guidelines. Therefore, the first three sections discuss the *Scope* of the standard, *Normative References* and *Definitions*, in that order.

CAUTION — ISA ADHERES TO THE POLICY OF THE AMERICAN NATIONAL STANDARDS INSTITUTE WITH REGARD TO PATENTS. IF ISA IS INFORMED OF AN EXISTING PATENT THAT IS REQUIRED FOR USE OF THE STANDARD, IT WILL REQUIRE THE OWNER OF THE PATENT TO EITHER GRANT A ROYALTY-FREE LICENSE FOR USE OF THE PATENT BY USERS COMPLYING WITH THE STANDARD OR A LICENSE ON REASONABLE TERMS AND CONDITIONS THAT ARE FREE FROM UNFAIR DISCRIMINATION.

EVEN IF ISA IS UNAWARE OF ANY PATENT COVERING THIS STANDARD, THE USER IS CAUTIONED THAT IMPLEMENTATION OF THE STANDARD MAY REQUIRE USE OF TECHNIQUES, PROCESSES, OR MATERIALS COVERED BY PATENT RIGHTS. ISA TAKES NO POSITION ON THE EXISTENCE OR VALIDITY OF ANY PATENT RIGHTS THAT MAY BE INVOLVED IN IMPLEMENTING THE STANDARD. ISA IS NOT RESPONSIBLE FOR IDENTIFYING ALL PATENTS THAT MAY REQUIRE A LICENSE BEFORE IMPLEMENTATION OF THE STANDARD OR FOR INVESTIGATING THE VALIDITY OR SCOPE OF ANY PATENTS BROUGHT TO ITS ATTENTION. THE USER SHOULD CAREFULLY INVESTIGATE RELEVANT PATENTS BEFORE USING THE STANDARD FOR THE USER'S INTENDED APPLICATION.

HOWEVER, ISA ASKS THAT ANYONE REVIEWING THIS STANDARD WHO IS AWARE OF ANY PATENTS THAT MAY IMPACT IMPLEMENTATION OF THE STANDARD NOTIFY THE ISA STANDARDS AND PRACTICES DEPARTMENT OF THE PATENT AND ITS OWNER. ADDITIONALLY, THE USE OF THIS STANDARD MAY INVOLVE HAZARDOUS MATERIALS, OPERATIONS OR EQUIPMENT. THE STANDARD CANNOT ANTICIPATE ALL POSSIBLE APPLICATIONS OR ADDRESS ALL POSSIBLE SAFETY ISSUES ASSOCIATED WITH USE IN HAZARDOUS CONDITIONS. THE USER OF THIS STANDARD MUST EXERCISE SOUND PROFESSIONAL JUDGMENT CONCERNING ITS USE AND APPLICABILITY UNDER THE USER'S PARTICULAR CIRCUMSTANCES. THE USER MUST ALSO CONSIDER THE

APPLICABILITY OF ANY GOVERNMENTAL REGULATORY LIMITATIONS AND ESTABLISHED SAFETY AND HEALTH PRACTICES BEFORE IMPLEMENTING THIS STANDARD.

THE USER OF THIS DOCUMENT SHOULD BE AWARE THAT THIS DOCUMENT MAY BE IMPACTED BY ELECTRONIC SECURITY ISSUES. THE COMMITTEE HAS NOT YET ADDRESSED THE POTENTIAL ISSUES IN THIS VERSION.

The following people served as active members in the revision of this document:

NAME	COMPANY
David Bell	ATR Distributing Co (Wonderware)
Jerry Golden	B&R Automation
John Kowal	B&R Automation
Marc Wolf	B&R Automation
Ryan Stell	B&R Automation
Carl Bostrom	Bosch Rexroth
Dave Chappell	CMAa
Arthur Smith	Corning
James Hoban	Eaton
Kevin Pitts	Eaton
Doug Buschor	E-Technologies
John Spengler	GE
Niels Andersen	Invensys
Kent Knudsen	Krones
Brian Leboeuf	Krones
Uwe Keiter	Lenze
Lee Smith	Mettler Toledo
Steven Abramowski	Miller Coors
Jerry Yen	Mitsubishi Electric
David Kaley	Mitsubishi Electric
Vili Taukolo	Mitsubishi Electric
Fabrice Bertin	Nestlé
Tom Doney	Nestlé
Bryan Griffen	Nestlé
Toshiya Kitagawa	Omron
Tatsuru Hyodo	Omron
Rick Schoonover	Patti Engineering
Jeff Russell	PepsiCo
Murugan Govindasamy	Pfizer
Eelco van der Wal	PLCopen
Tom Egan	PMMI
Mark Ruberg	Pro Mach
Christopher Thomas	Pro Mach (Axon)
Gord Davison	Pro Mach (Edson)
Jason DeBruler	Procter & Gamble
Dan Amundson	Procter & Gamble
John Dart	Rockwell Automation
Dan Seger	Rockwell Automation
Yann Renard	Sidel
Stephane Hacpille	Sidel
Mike Pieper	Siemens Industry
Rana, Ajay S	Siemens Industry
Alexander Stukenkemper	Siemens Industry
Oliver Welle	Siemens AG
Dewayne A Bruschi	SMC
R Johnson	SMC
Doug Meyer	Yaskawa

The following people served as active members in the initial creation of this document:

NAME	COMPANY
Joseph Jablonski	Acumence Inc
Barry Kluener	Alexander & Associates
David Bell	ATR Distribution (Wonderware)
Thomas Hopfgartner	B&R Automation
Uwe Keiter	B&R Automation
Gerd Hoppe	Beckhoff
David Arens	Bosch Rexroth
Dennis Brandl	BR&L Consulting
David Chappell	CMAa-LLC
Joe Faust	Douglas Machine Company
Dominik Gludowatz	Elau
Tom Jensen	Elau
Andre Uhl	Elau
David Bauman	ISA / OMAC
Alex Pereira	KHS
Randy Dwiggin	Maverick Technologies
Ron MacDonald	Nestlé
Eelco VanDerWal	PLCopen
Mike Lamping	Procter & Gamble
Ulrich Arlt	Rockwell Automation
Darren Elliott	Rockwell Automation
Brian Hedges	Rockwell Automation
John Dart	Rockwell Automation
Paul Nowicki	Rockwell Automation
Dan Seger	Rockwell Automation
Garth Basson	SAB Miller
Larry Trunek	SAB Miller
Willie Lotz	SAB Miller Brewing Co.
Fabian Ochoa M.	SAB Miller Brewing Co.
Mario Broucke	Siemens AG, A&D
Roland Heymann	Siemens AG, A&D
Detlef Rausch	Siemens AG, A&D
Robert Freller	Siemens AG, F&B
Dr. Holger Grzonka	Siemens Energy & Automation
Mike Pieper	Siemens Energy & Automation
Mark DeCramer	WAGO
Dr. Tobias Voigt	Weihenstephan University

This document was approved for publication by the ISA Standards and Practices Board on 1 July 2015.

NAME	AFFILIATION
N. Sands, Vice President	DuPont
D. Bartusiak	ExxonMobil Research & Engineering
P. Brett	Honeywell Inc.
E. Cosman	OIT Concepts, LLC
K. Demachi	Yokogawa Electric Corp.
B. Dumortier	Schneider Electric
D. Dunn	Consultant
J. Federlein	Federlein & Assoc. Inc.
B. Fitzpatrick	Wood Group Mustang

J. Gilsinn
J.-P. Hauet
E. Icaan
J. Jamison
K. P. Lindner
V. Maggioli
T. McAviney
V. Mezzano
C. Monchinski
H. Sasajima
T. Schnaare
J. Tatera
K. Unger
I. Verhappen
W. Weidman
J. Weiss
M. Wilkins
D. Zetterberg

Kenexis Consulting
KB Intelligence
Atkins
Encana Corp.
Endress + Hauser Process Solutions AG
Feltronics Corp.
Instrumentation and Control Engineering, LLC
Fluor Corporation
Automated Control Concepts Inc.
Azbil Corp.
Rosemount Inc.
Tatera & Associates Inc.
Stone Technologies Inc.
Orbis Engineering Field Services
WCW Consulting
Applied Control Solutions LLC
Yokogawa IA Global Marketing (USMK)
Chevron Energy Technology Co.

CONTENTS

Foreword.....	11
Abstract	11
Key words	11
1 Scope	13
2 References	13
3 Overview.....	14
3.1 Introduction.....	14
3.2 Personnel and environmental protection	15
3.3 Control system compatibility	15
4 Unit/Machine states.....	15
4.1 Definition	15
4.2 Types of states.....	16
4.3 Defined states.....	16
4.4 State transitions and state commands.....	19
4.4.1 Definition.....	19
4.4.2 7.5.2 Types of state commands.....	19
4.4.3 Examples of state transitions	19
4.5 State model.....	22
4.5.1 Base state model.....	22
5 Modes.....	23
5.1 Unit/Machine control modes	23
5.2 Unit/Machine control mode management.....	24
6 Common unit/machine mode examples	26
6.1 Production mode	26
6.2 Maintenance mode	26
6.3 Manual mode	28
6.4 User definable modes.....	30
7 Automated machine functional tag description.....	31
7.1 Introduction to PackTags	31
7.2 Tag types	31
7.3 PackTags names.....	32
7.4 Data types, units, and ranges	32
7.4.1 Structured data types.....	32
7.5 Tag details	33
7.5.1 Command tags	45
7.5.1.1 Command.UnitMode	45
7.5.1.2 Command.UnitModeChangeRequest.....	45
7.5.1.3 Command.MachSpeed.....	46
7.5.1.4 Command.MaterialInterlock	46
7.5.1.5 Command.CntrlCmd	47

7.5.1.6	Command.CmdChangeRequest	47
7.5.1.7	Command.RemoteInterface[#]	47
7.5.1.8	Command.Parameter[#]	49
7.5.1.9	Command.Product[#]	50
7.5.2	Status tags	53
7.5.2.1	Status.UnitModeCurrent	53
7.5.2.2	Status.UnitModeRequested	53
7.5.2.3	Status.UnitModeChangeInProgress	53
7.5.2.4	Status.StateCurrent	54
7.5.2.5	Status.StateRequested	54
7.5.2.6	Status.StateChangeInProgress	55
7.5.2.7	Status.MachSpeed	55
7.5.2.8	Status.CurMachSpeed	55
7.5.2.9	Status.MaterialInterlock	56
7.5.2.10	Status.EquipmentInterlock.Blocked	56
7.5.2.11	Status.EquipmentInterlock.Starved	56
7.5.2.12	Status.RemoteInterface[#]	57
7.5.2.13	Status.Parameter[#]	58
7.5.2.14	Status.Product[#]	59
7.5.3	Administration tags	62
7.5.3.1	Admin.Parameter[#]	62
7.5.3.2	Admin.Alarm[#]	63
7.5.3.3	Admin.AlarmExtent	65
7.5.3.4	Admin.AlarmHistory[#]	65
7.5.3.5	Admin.AlarmHistoryExtent	66
7.5.3.6	Admin.StopReason[#]	66
7.5.3.7	Admin.StopReasonExtent	68
7.5.3.8	Admin.Warning[#]	68
7.5.3.9	Admin.WarningExtent	69
7.5.3.10	Admin.ModeCurrentTime[#]	69
7.5.3.11	Admin.ModeCumulativeTime[#]	69
7.5.3.12	Admin.StateCurrentTime[#,#]	69
7.5.3.13	Admin.StateCumulativeTime[#,#]	69
7.5.3.14	Admin.ProdConsumedCount[#]	70
7.5.3.15	Admin.ProdProcessedCount[#]	71
7.5.3.16	Admin.ProdDefectiveCount[#]	72
7.5.3.17	Admin.AccTimeSinceReset	74
7.5.3.18	Admin.MachDesignSpeed	74
7.5.3.19	Admin.StatesDisabled	74
7.5.3.20	Admin.PLCDateTime	74
Annex A	— Weihenstephan harmonization	75
Annex B (informative)	— Overview of 2015 Technical Report changes	81

LIST OF FIGURES AND TABLES

Figure 1: Automated machines applied to ANSI/ISA-88.00.01 physical model	14
Figure 2: Example states for automated machines	16
Table 1 : Complete list of machine dtates	17
Table 2 : Example transition matrix of local or remote state commands	20
Table 3: Example matrix of machine conditions initiating a state command	21
Figure 3: Base state model visualization	22
Figure 4: Unit/Machine control modes	23
Figure 5: Multi-mode example diagram (production mode states)	25
Figure 6: Maintenance mode state model	26
Figure 7: Maintenance mode execution model	27
Figure 8: Manual mode state model	28
Figure 9: Manual mode execution model	29
Figure 10: Automatic Weihenstephan state model	30
Table 4: Command tags (complete listing)	34
Table 5: Status tags (complete listing)	36
Table 6: Admin tags (complete listing)	38
Table 7: PackTags: Minimum required for information/machine monitoring	43
Table 8: PackTags: Minimum required for supervisory control.....	44
Figure 12: Unit mode change example sequence.....	45
Figure 13: Unit mode change example sequence.....	54
Figure 14: State change example sequence	55
Table A1 - Weihenstephan parameterized control tags	75
Table A2 - Weihenstephan parameterized status tags	76
Table A3 - Weihenstephan parameterized administration tags	79

This page intentionally left blank.

Foreword

The ISA88 committee has defined a series of standards addressing the batch industry and providing terminology and a consistent set of concepts and models for batch manufacturing plants and batch control. These standards, however, were not defined in the context of packaging machines, or machines that perform discrete operations. As the ANSI/ISA88 batch standard continues to evolve, the context of the standard models may be extended to include the entire plant, integrating the software definitions of batch, packaging, converting and warehousing. Currently, as noted in this report there is a need to begin consideration of the ANSI/ISA88 standard in the context of differing automated machinery.

This is an informative document. This document contains implementation guidelines in order to establish a common presentation and high level software architecture or layout. The terms and definitions used in this document are harmonized, as much as possible, with ANSI/ISA-88.00.01; the document is not definitive in this respect. The models used, and applied, in this document are an extension of the models presented in ANSI/ISA-88.00.01 and are shown how they are applied to differing machine functionality. Discrete machine functionality is expressed graphically in several situations and described. The intent of this document is proposing specific implementation options and indicates a preference for a specific set of machine types.

In 2013 this document was updated for three reasons: to simplify the document and enable easier adoption, to clarify existing materials and make them easier to apply and to make it more complete, including the best approaches being used to implement. Major changes include addition of a minimum set of PackTags, a minimum set of states and removal of examples and MES definitions not central to the document's purpose. Other changes addressed in this revision are: improved definition of suspending and holding, transition between modes, blocked and starved tags, stop reasons, and warnings.

Publication of this technical report that has been registered with ANSI has been approved by ISA, 67 Alexander Drive, Research Triangle Park, NC 27709. This document is registered as a technical report according to the Procedures for the Registration of Technical Reports with ANSI. This document is not an American National Standard and the material contained herein is not normative in nature. Comments on the content of this document should be sent to ISA, 67 Alexander Drive, Research Triangle Park, NC 27709.

Abstract

The approaches used in programming discrete machines today are generally considered to be solely dependent on the machine and the software engineer, or control systems programmer. This constant change offers little additional value and generally increases the total costs, from the designing and building of the process to operating and maintaining the system by the end user. This technical report on the implementation of ANSI/ISA-88.00.01 in discrete machines breaks this paradigm and demonstrates how to apply the ANSI/ISA-88.00.01 standard concepts to automated machine states and modes. This technical report gives examples of general and specific machine state models and procedural methods. The report cites real control examples as implementations, and provides specific tag naming conventions; it also cites a number of common terms that are consistent with batch processing and ANSI/ISA-88.00.01.

Key words

state machine, state model, mode manager, machine state, unit control mode, PackML, state commands, command tags, status tags, administration tags, base state model, functional programming, modular programming, machine control software, discrete machine software, PackTags, Weihenstephan, Production Data Acquisition, PDA, ISA88, TR88.

Introduction

When the ANSI/ISA-88.00.01 standard is applied to applications across a plant, there is a need to align the terminologies, models and key definitions between different process types: continuous, batch, and discrete processes. Discrete processes involve machines found in the packaging, converting, and material handling applications. The operation of these machines is typically defined by the OEM, system integrator, end user, or is industry specific.

A task group with members from technology providers, OEMs, system integrators, and end users was chartered by the OMAC (Organization for Machine Automation and Control) Packaging Workgroup. The task group generated the PackML guidelines as a method to show how the ANSI/ISA-88.00.01 concepts could be extended into packaging machinery. This technical report is intended to build upon and formalize the concepts of the PackML guidelines and to show application examples.

The purpose of the technical report is to:

- Define a standard state-based model for automated machines.
- Identify definitions for common terminology.
- Explain to practitioners how to use state programming for automated machines.
- Provide references to actual implementation examples and templates from automation and control vendors.
- Identify a common tag structure for automated machines in order to:
 - Provide for “connect & pack” functionality
 - Provide functional interoperability and a consistent look and feel across the plant floor.
 - Provide consistent tag structure for connection to plant MES and enterprise systems.

Machine and Unit States ANSI/ISA-TR88.00.02-2015

1 Scope

Since its inception, the OMAC Packaging Machine Language (PackML) group has been using a variety of information sources and technical documents to define a common approach, or machine language, for automated machines. The primary goals are to encourage a common “look and feel” across a plant floor, and to enable and encourage industry innovation. The PackML group is recognized globally and consists of control vendors, OEM’s, system integrators, universities, and end users, which collaborate on definitions that endeavour to be consistent with the ISA88 standards and consistent with the technology and the changing needs of a majority of automated machinery. The term “machine” used in this report is equivalent to an ISA88 “unit”.

This has led to the following:

1. A definition of machine/unit state types
2. A definition of machine/unit control modes
3. A definition of unit control mode management
4. State models, state descriptions, and mode and state transitions
5. A definition of the minimum PackTags required for performance monitoring

2 References

The following documents contain provisions that are referenced in this text. At the time of publication the editions indicated were valid. All documents are subject to revision, and parties to agreements based on this technical report are encouraged to investigate the possibility of applying the most recent editions of the reference documents indicated below.

- ANSI/ISA-88.00.01-2010, Batch Control - Part 1: Models and Terminologies
- ISA-88.00.02-2001, Batch Control - Part 2: Data Structures and Guidelines for Languages
- ANSI/ISA-88.00.03-2003, Batch Control - Part 3: General and Site Recipe Models and Representation
- ANSI/ISA-88.00.04-2006, Batch Control - Part 4: Batch Production Records
- ISA Draft 88.00.05 Batch Control - Part 5: Implementation Models & Terminology for Modular Equipment Control
- IEC 61131-1 Programmable controllers – Part 1: General Information
- IEC 61131-3 Programmable controllers - Part 3: Programming Languages
- IEC TR61131-4 Programmable controllers - Part 4: User Guidelines
- PLCopen TC5 Safety Software
- Weihenstephan Standard – Part 2 Version 2005 www.weihenstephaner-standards.de/index.php?id=2&L=1
- ANSI/ISA-95.00.01-2010 (IEC 62264-1 Mod) Enterprise – Control System Integration - Part 1: Models and Terminologies
- ANSI/ISA-95.00.02-2010 (IEC 62264-2 Mod) Enterprise – Control System Integration - Part 2: Object Model Attributes
- ANSI/ISA-95.00.05-2013, Enterprise - Control System Integration - Part 5: Business-to-Manufacturing Transactions
- DIN 8782, Beverage Packaging Technology; Terminology Associated with Filling Plants and their Constituent Machines

- www.omac.org – Organization for Machine Automation and Control Website
- ISA-TR88.00.02-2008 – The original edition of this technical report.
- ISO 22400 Automation Systems and Integration – Key performance indicators (KPIs) for manufacturing operations management.

3 Overview

3.1 Introduction

Automated machine programming is typically done by software engineers, machine designers, and system integrators. The form and style of the machine software ranges from modular, to monolithic in nature. The objective of this report is to specify the application of a common software methodology that is consistent with the modular programming of automated machinery as described in the draft ISA-88.00.05 standard. The naming of specific software components, or operational aspects, is dependent on the needs of the automated machine. This report shall be interpreted in a general sense to encompass all automated machinery. It is focused on the overall operation and functionality of automated machines. This document enables a consistent method of machine interconnection and operability. The diagrams and examples shown in the report are specific in terms of the functionality they provide but can be implemented in various ways to fit most automated machinery and machine controllers; therefore the figures do not follow ISO/IEC 19501:2005 for depiction of software flow.

If automated machinery is modelled in an ANSI/ISA-88.00.01 physical hierarchy, the example mapping shown in Figure 1 is possible. The example in this document will assume that a machine can represent the unit level in the ISA88 hierarchy.

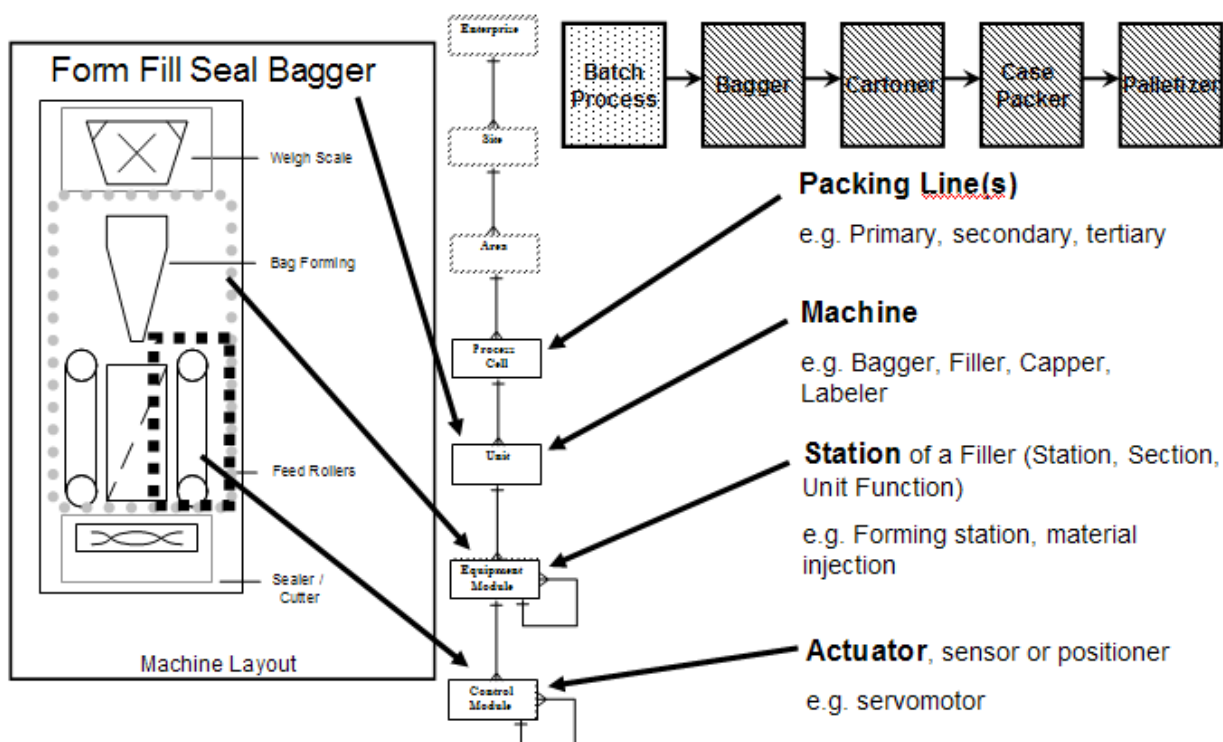


Figure 1: Automated machines applied to ANSI/ISA-88.00.01 physical model

Furthermore, the objective of this document is to provide a definition of machine (unit) modes and states, as well as state models corresponding to the different machine (unit) modes. In this example application of ANIS/ISA-88.00.01, the use of the state model and unit modes are extensible, but the methods governing the way in which the modes and states are used is not. This technical report demonstrates the flexibility and ease in which this method can be implemented in terms of ISA88, as well as how it provides the “common look and feel” desired in automated machines. This model only constrains the standard names and semantics for commonly used high level machine states as per a Base State Model.

The ANSI/ISA-88.00.01 standard describes example modes and states as applied to equipment entities and procedural elements. This report identifies unit/machine modes and states which should be considered an extension of the examples in the ANSI/ISA-88.00.01 standard in order to meet the needs of automated machine processing.

3.2 Personnel and environmental protection

The personnel and environmental protection control activity provides safety for people and the environment. No control activity should intervene between personnel and environmental protection and the field hardware it is designed to operate with. Personnel and environmental protection is, by definition, separate from the higher level control activities in this document. It may map to more than one software level of the equipment as desired.

A complete discussion of personnel and environmental protection, the classification of these types of systems, and the segregation of levels of interlocks within these systems is a topic of its own and beyond the scope of this document.

3.3 Control system compatibility

The minimum PackTag requirements that are identified in this document can be executed on a majority of PLC systems used by packaging machine manufacturers. It can also be implemented on systems such as check weighers, date coders, robotics, and other ancillary devices that may utilize proprietary control platforms. Even though the PackTags naming convention is using logical names, it can be implemented using register memory systems.

4 Unit/Machine states

4.1 Definition

A unit/machine state completely defines the current condition of a machine. A machine state is determined by an ordered procedure, or programming routine, that can consist of one or more commands to other procedural elements¹ or equipment entities, or be affected by the status of a procedural element¹ or equipment entity, or both. In performing the function specified by the state, the machine software will issue a set of commands to the machine procedural elements¹ or equipment entities which in turn can report status.

Only one major processing activity may be active in one machine at any time². The linear sequence of major activities will drive a strictly sequentially ordered flow of control from one state to the next state – no more than one state of the base model is allowed to be active in one machine at the same time.

¹ Term “procedural element” defined (ANSI/ISA-88.00.01)

² A “major processing activity” corresponds to the term “equipment operation” as defined in ANSI/ISA-88.00.01.

The 2010 revision to this technical report added a minimum set of states in Figure 2. It also clarified the difference between holding and suspending in Table 1.

Note: At a lower level, the minor sub-activities (or control procedures) that are combined to form a major activity at the machine operation level, may indeed be taking place in parallel as well as in sequence as defined in ANSI/ISA-88.00.01 for equipment phases.

4.2 Types of states

For the purposes of understanding, two machine state types are defined:

- **Acting state:** A state which represents some processing activity. It implies the single or repeated execution of processing steps in a logical order, for a finite time or until a specific condition has been reached. In ANSI/ISA-88.00.01 these are referred to transient states, those states ending in “ING”.
- **Wait state:** A state used to identify that a machine has achieved a defined set of conditions. In such a state, the machine is maintaining a status until transitioning to an acting state. In ANSI/ISA-88.00.01 this was referred to as a “final” or “quiescent” state.

4.3 Defined states

There are a fixed number of states defined in the base state model. This report establishes an example enumerated set of possible unit/machine states illustrated in the figure below, Figure 2. As shown, this set of states is a subset of the ANSI/ISA-88.00.01-2010 example states. Several states defined there are not supported for machine processing. Also see Figure 3.

Note: This table has been updated to show the “minimum required” states for a machine to be considered an appropriate PackML implementation.

ANSI/ISA-88.00.01 Procedural States	ISA-TR88.00.02 Equipment States				
	Value	Unit / Machine States	Wait	Acting	Minimum Required
<not defined>	1	Clearing		x	
Stopped	2	Stopped	x		x
<not defined>	3	Starting		x	
Idle	4	Idle	x		x
Paused	5	Suspended	x		
Running	6	Execute	x	x	x
Stopping	7	Stopping		x	
Aborting	8	Aborting		x	
Aborted	9	Aborted	x		x
Holding	10	Holding		x	
Held	11	Held	x		
Restarting	12	Unholding		x	
Pausing	13	Suspending		x	
<not defined>	14	Unsuspending		x	
<not defined>	15	Resetting		x	
<not defined>	16	Completing		x	
Complete	17	Complete	x		

Figure 2: Example states for automated machines

Formal definitions of these states are given below:

Table 1 : Complete list of machine states

State Name	Description
STOPPED	State Type: Wait The machine is powered and stationary after completing the STOPPING state. All communications with other systems are functioning (if applicable). A <i>RESET</i> command will cause an exit from STOPPED to the RESETTING state.
STARTING	State Type: Acting The machine completes the steps needed to start. This state is entered as a result of a <i>STARTING</i> command (local or remote). Following this command the machine will begin to "execute".
IDLE	State Type: Wait This is the state which indicates that RESETTING is complete. The machine will maintain the conditions which were achieved during the RESETTING state, and perform operations required when the machine is in IDLE.
SUSPENDING	State Type: Acting This state shall be used when EXTERNAL (outside this unit machine but usually on the same integrated production line) process conditions do not allow the machine to continue producing, that is, the machine leaves EXECUTE due to upstream or downstream conditions on the line. This is typically due to a blocked or starved event. This condition may be detected by a local machine sensor or based on a supervisory system external command. While in the SUSPENDING state, the machine is typically brought to a controlled stop and then transitions to SUSPENDED upon state complete. To be able to restart production correctly after the SUSPENDED state, all relevant process set-points and return status of the procedures at the time of receiving the <i>SUSPEND</i> command must be saved in the machine controller when executing the SUSPENDING procedure.
SUSPENDED	State Type: Wait Refer to SUSPENDING for when this state is used. In this state the machine shall not produce product. It will either stop running or continue to cycle without producing until external process conditions return to normal, at which time, the SUSPENDED state will transition to the UNSUSPENDING state, typically without any operator intervention.
UNSUSPENDING	State Type: Acting Refer to SUSPENDING for when this state is used. This state is a result of process conditions returning to normal. The UNSUSPENDING state initiates any required actions or sequences necessary to transition the machine from SUSPENDED back to EXECUTE. To be able to restart production correctly after the SUSPENDED state, all relevant process set-points and return status of the procedures at the time of receiving the <i>SUSPEND</i> command must be saved in the machine controller when executing the SUSPENDING procedure.
EXECUTE	State Type: Acting Once the machine is processing materials it is in the EXECUTE state. Different machine modes will result in specific types of EXECUTE activities. For example, if the machine is in the "Production" mode, the EXECUTE will result in products being produced, while in "Clean Out" mode the EXECUTE state refers to the action of cleaning the machine.

State Name	Description
STOPPING	State Type: Acting This state is entered in response to a <i>STOP</i> command. While in this state the machine executes the logic which brings it to a controlled stop as reflected by the STOPPED state. Normal STARTING of the machine cannot be initiated unless RESETING had taken place.
ABORTING	State Type: Acting The ABORTING state can be entered at any time in response to the <i>ABORT</i> command or on the occurrence of a machine fault. The aborting logic will bring the machine to a rapid safe stop.
ABORTED	State Type: Wait The machine maintains status information relevant to the ABORT condition. The machine can only exit the ABORTED state after an explicit <i>CLEAR</i> command, subsequently to manual intervention to correct and reset the detected machine faults.
HOLDING	State Type: Acting This state shall be used when INTERNAL (inside this unit machine and not from another machine on the production line) machine conditions do not allow the machine to continue producing, that is, the machine leaves EXECUTE due to internal conditions. This is typically used for routine machine conditions that requires minor operator servicing to continue production. This state can be initiated automatically or by an operator and can be easily recovered from. An example of this would be a machine that requires an operator to periodically refill a glue dispenser or carton magazine and due to the machine design, these operations cannot be performed while the machine is running. Since these types of tasks are normal production operations, it is not desirable to go through aborting or stopping sequences, and because these functions are integral to the machine they are not considered to be "external". While in the HOLDING state, the machine is typically brought to a controlled stop and then transitions to HELD upon state complete. To be able to restart production correctly after the HELD state, all relevant process set-points and return status of the procedures at the time of receiving the <i>HOLD</i> command must be saved in the machine controller when executing the HOLDING procedure.
HELD	State Type: Wait Refer to HOLDING for when this state is used. In this state the machine shall not produce product. It will either stop running or continue to dry cycle. A transition to the UNHOLDING state will occur when INTERNAL machine conditions change or an <i>UNHOLD</i> command is initiated by an operator.
UNHOLDING	State Type: Acting Refer to HOLDING for when this state is used. A machine will typically enter into UNHOLDING automatically when INTERNAL conditions, material levels, for example, return to an acceptable level. If an operator is required to perform minor servicing to replenish materials or make adjustments, then the <i>UNHOLD</i> command may be initiated by the operator.
COMPLETING	State Type: Acting This state is an automatic response from the EXECUTE state. Normal operation has run to completion, i.e. processing of material at the infeed will stop.
COMPLETE	State Type: Wait The machine has finished the COMPLETING state and is now waiting for a <i>RESET</i> command before transitioning to the RESETING state.
RESETING	State Type: Acting This state is the result of a <i>RESET</i> command from the STOPPED or COMPLETE state. Faults and stop causes are reset. RESETING will typically cause safety devices to be energized and place the machine in the IDLE state where it will wait for a <i>START</i> command. No hazardous motion should happen in this state.

State Name	Description
CLEARING	State Type: Acting Initiated by a state command to clear faults that may have occurred when ABORTING, and are present in the ABORTED state before proceeding to a STOPPED state.

4.4 State transitions and state commands

4.4.1 Definition

A state transition is defined as a passage from one state to another. Transitions between states will occur as a result of a local, remote, or procedural state command. State commands are procedural elements that in effect cause a state transition to occur.

4.4.2 7.5.2 Types of state commands

State commands are comprised of one or a combination of the following types:

- Operator intervention.
- Response to the status of one or more procedural elements.
- Response to machine conditions.
- The completion of an acting state procedure.
- Supervisory or remote system intervention.

4.4.3 Examples of state transitions

An example transition matrix for local or remote state commands generated by an operator is shown in Table 2. After every acting state, as can be seen in Table 2, a procedural element is required that will indicate the acting state is complete, or a command is required to stop or abort the acting state. The state complete indication within the acting state procedure will cause a state transition to occur.

Similarly, an example state command matrix of machine conditions activating a state command is shown in Table 3. The objective of this table is to depict the machine conditions that will cause a state transition using the commands defined in Table 2.

Table 2 : Example transition matrix of local or remote state commands

Current State	State Commands									State Complete
	Start	Reset ¹	Hold	Un-Hold	Suspend	UnSuspended	Clear ¹	Stop ²	Abort ²	
IDLE	STARTING							STOPPING	ABORTING	
STARTING								STOPPING	ABORTING	EXECUTE
EXECUTE			HOLDING		SUSPENDING			STOPPING	ABORTING	COMPLETING
COMPLETING								STOPPING	ABORTING	COMPLETE
COMPLETE		RESETTING						STOPPING	ABORTING	
RESETTING								STOPPING	ABORTING	IDLE
HOLDING								STOPPING	ABORTING	HELD
HELD				UNHOLDING				STOPPING	ABORTING	
UNHOLDING								STOPPING	ABORTING	EXECUTE
SUSPENDING								STOPPING	ABORTING	SUSPENDED
SUSPENDED						UNSUSPENDING		STOPPING	ABORTING	
UNSUSPENDING								STOPPING	ABORTING	EXECUTE
STOPPING									ABORTING	STOPPED
STOPPED		RESETTING							ABORTING	
ABORTING										ABORTED
ABORTED							CLEARING			
CLEARING									ABORTING	STOPPED

¹ It is common practice for clearing and resetting COMMANDS to be initiated using the same physical operator interface device.

² It is common practice in packaging (but not process) applications to permit use of STOP and ABORT commands while in the IDLE, COMPLETE, STOPPED, and RESETTNG states.

Table 3: Example matrix of machine conditions initiating a state command

	Example Machine Conditions										
Current State	Operat or Start	Carton Magazi ne Low	Carton Magazi ne Full	Downstre am Not Ready	Downstre am Ready	E- Stop or Fault	No Produ ct Present	Product Present	Operat or Stop	Produc t Count Reache d	Clea r Faults
IDLE	Start					Abor t			Stop		
STARTING						Abor t			Stop		
EXECUTE		Hold		Suspend		Abor t	Suspe nd		Stop	Compl ete	
COMPLETING						Abor t			Stop		
COMPLETE						Abor t			Stop		
RESETTING						Abor t			Stop		
HOLDING						Abor t			Stop		
HELD			UnHold			Abor t			Stop		
UNHOLDING						Abor t			Stop		
SUSPENDING						Abor t			Stop		
SUSPENDED					UnSuspe nd	Abor t		UnSuspen ding	Stop		
UNSUSPENDING						Abor t			Stop		
STOPPING						Abor t					
STOPPED						Abor t					
ABORTING											
ABORTED											Clea r
CLEARING						Abor t					

4.5 State model

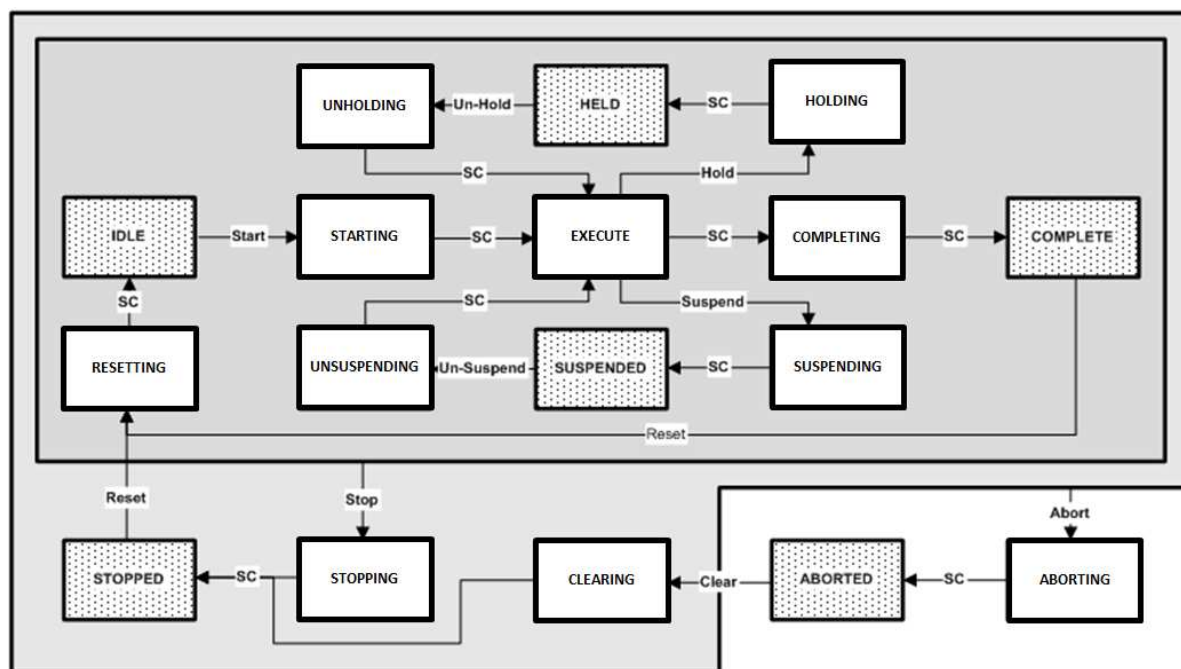
A state model completely defines the behaviour of a machine in one unit control mode. In a state model, states are arranged in an ordered fashion that is consistent with the specified operation of the machine. The specific states required are dependent on the machine operation.

The compilation of all defined functional states and their transitions is called the base state model.

4.5.1 Base state model

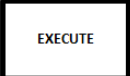
The base state model represents the complete set of defined states, state commands, and state transitions. All unit control modes will be defined by subsets of the base state model. The base state model is depicted in Figure 3. This model is a subset of the state model from ANSI/ISA-88.00.01-2010 (Annex D Figure 29) with three distinctions:

- The paused or pausing states are not used
- The label "execute" is used for the running state
- Transition to holding directly from suspending, suspend or unsuspending is not supported



SC = State Complete

 = Wait State

 = Acting State

States within the dark gray outline can transition to the STOPPING state or the ABORTING state.
States within the light gray outlined can transition to ABORTING state.
The minimum states are identified in Figure 2.

Figure 3: Base state model visualization

5 Modes

The ANSI/ISA-88.00.01 standard provides a set of modes for equipment entities and procedural elements. This report establishes modes for automated machines that are considered different than the ANSI/ISA-88.00.01 procedural modes (automatic, semi-automatic, and manual). The ANSI/ISA-88.00.01 procedural modes describe the way procedures operate. Procedural modes are not common in automated machinery. They are provided for in this report by noting they may exist, but are not considered in the scope of this work and are not included in the tag table.

For automated machines, the example in this report establishes unit/machine modes in order to allow a machine designer to adjust the set of states, state commands and state transitions a machine may follow given different operating circumstances. The set of defined unit/machine control modes is shown in Figure 4 and described below.

The 2013 revision to this document changed producing mode to production mode.

ANSI/ISA-88.00.01 Example Modes	ISA-TR88.00.02 Control Modes	
	Value	Unit / Machine Control Modes
<not defined>	0	Invalid
<not defined>	1	Production
<not defined>	2	Maintenance
<not defined>	3	Manual
<not defined>	04 – 31	User Definable

Figure 4: Unit/Machine control modes

5.1 Unit/Machine control modes

A unit/mMachine control mode determines the subset of states, state commands, and state transitions that determine the strategy for carrying out a unit/machine's process.

Typical unit/machine control modes are production, maintenance, manual, clean in place, run out, semi-auto, dry cycle, etc. The distinguishing elements between these unit control modes are the selected subset of states, state commands, and state transitions.

The ordered procedures that control the states will be unique for the unit/machine control mode that the unit/machine is in. For example, in a "production" unit/machine control mode the definition of "execute" in a filling machine will mean it is "producing" product. In the "manual" unit/machine control mode the definition of the "executing" state may be jogging or indexing. The "execute" state defines the functional operation of the unit/machine control mode. States of identical names may have different functions in different unit/machine control modes

Examples of unit/machine control modes are:

Production mode - This represents the mode which is utilized for routine production. The machine executes relevant logic in response to commands which are either entered directly by the operator or issued by another supervisory system.

Maintenance mode - This mode may allow suitably authorized personnel the ability to run an individual machine independent of other machines in a production line. This mode would typically be used for fault finding, machine trials or testing operational improvements. This mode would also allow the speed of the machine to be adjusted (where this feature is available).

Manual mode - This provides direct control of individual machine modules. This feature is available depending upon the mechanical constraints of the mechanisms being exercised. This feature may be used for the commissioning of individual drives, verifying the operation of synchronized drives, testing the drive as a result of modifying parameters etc.

5.2 Unit/Machine control mode management

Automated machinery has unit/machine control modes other than “production”, as noted earlier. Each unit/machine control mode has its own state model. In order to manage the change from one mode to the next, a method of mode management must be defined. The mode management method determines how, and in what state a machine may change unit/machine control modes; i.e. the mode management method includes interlocks that prevent the machine changing unit/machine control modes when in inappropriate states.

Unit/machine control mode management enables the machine designer to manage unit/machine control mode transitions. Specification on transitions between unit/machine control modes is left to the user, but typical transition points are at *wait* states. The specification of the unit/machine control mode manager is such that no state or control functions are carried out in this upper level routine. The intent of the mode manager is to logically supervise when a change in mode can be done, command a mode change, and report status of the change request. All considerations of a mode manager must be consistent with prevailing safe practices and standards.

Transitions between unit/machine control modes can occur at only pre-programmed states:

- As a result of a local or remote operator command.
- As a result of a remote request from another automated unit.
- As a result of a state change. This is generated by change of state of one or a number of machine conditions, either directly from I/O or completion of a logic method. For example, if a filling machine has completed its production run in “PRODUCTION” mode of a given number of cases it may change to the “CLEANING” mode to begin a clean cycle.
- When a unit/machine control mode change takes place, a machine state change is NOT allowed to occur at the same time. Mode transitions must take place in a state that is common to both modes. This is necessary to avoid unintended machine sequences from taking place.

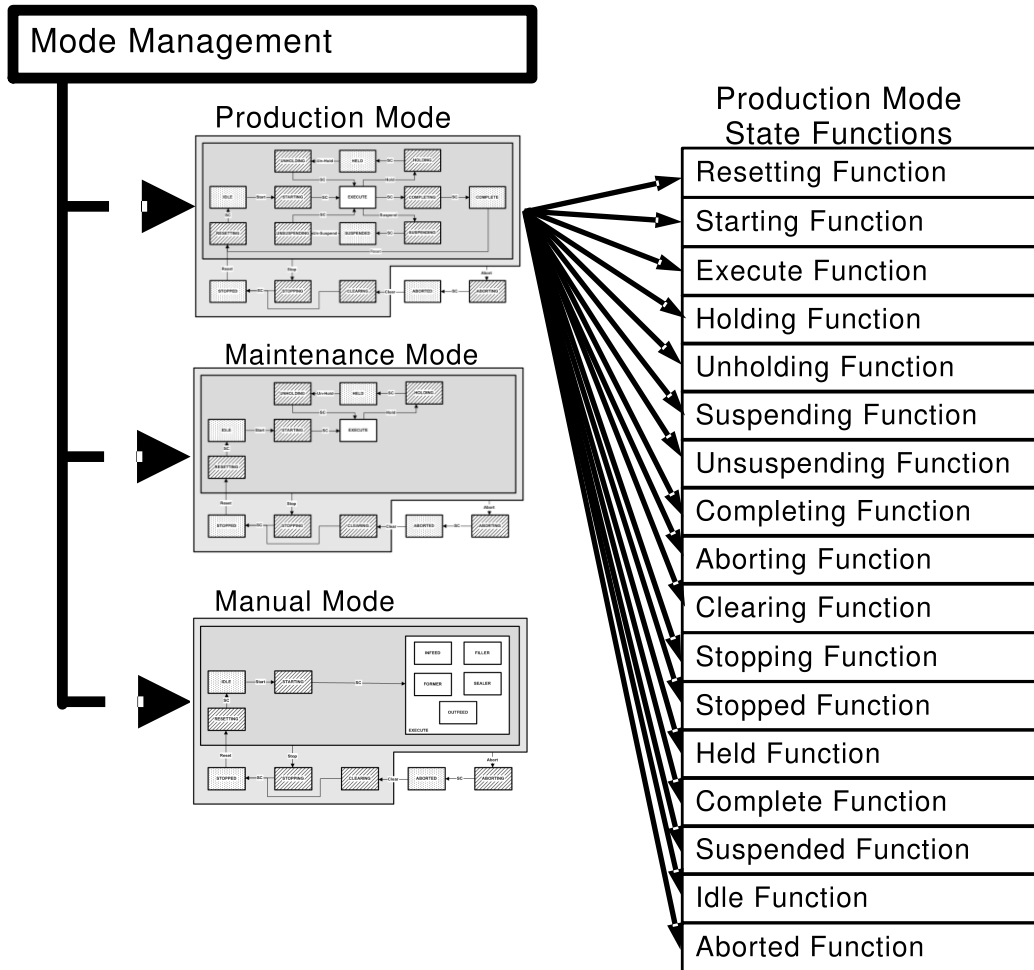


Figure 5: Multi-mode example diagram (production mode states)

6 Common unit/machine mode examples

6.1 Production mode

The base state model above can be used to define a production mode that is used in order to deliver control of routine processing and production. It is recognized that machines also require maintenance, calibration and setting up. To address this requirement two example modes of operation are shown below: maintenance and manual. Because there can be any number of possible modes for an automated machine, a user mode example is also shown. The user mode example is based on the Weihenstephan PDA (production data acquisition) standard.

6.2 Maintenance mode

Maintenance mode allows suitably authorized personnel the ability to run an individual machine independent of other machines in a production line. This would typically be used for fault finding, machine trials or testing operational improvements. It is expected that, because the machine will generally operate in its usual manner, it will need to undergo some or all of its routine starting up procedures. Maintenance mode will follow a recognized unit state model.

By way of example, one possible maintenance mode state model is shown in Figure 6 below. It is recognized that individual machine manufacturers may have good reason to develop other versions of maintenance mode state models. Typically modes, such as maintenance mode are developed as containing a subset of the unit states in the base state model. The unit state names remain consistent but the function of the states has been modified to be consistent with the mode function.

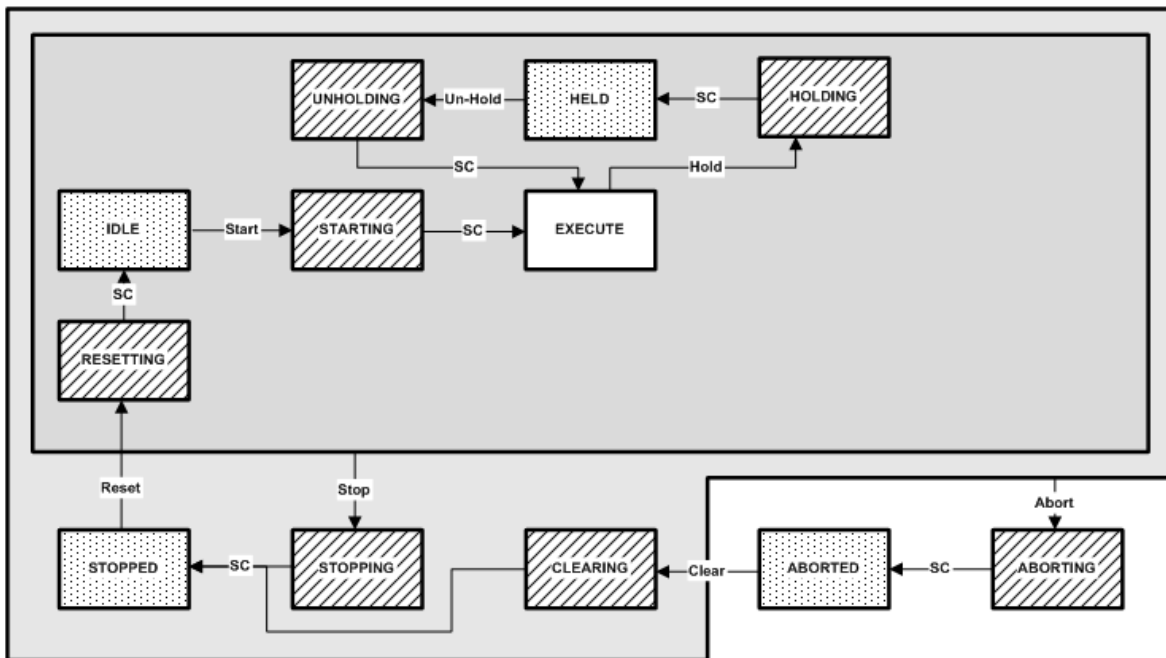


Figure 6: Maintenance mode state model

As can be seen above, the unit state model proposed for maintenance operations is a subset of the previously defined base state model. The essential difference between the base state model and maintenance mode state model is the absence of a SUSPENDED state in maintenance mode. It is envisaged for certain line types that the SUSPENDED state is not required, as its function is to provide for a wait state for incoming material. In this example maintenance mode is not designed for routine production and hence no SUSPENDED state is available. The function of EXECUTE

state has also taken on new meaning, in that EXECUTING production may not require the same logic as EXECUTING in maintenance.

In the figure below, the maintenance mode execution model is shown. Only unit state functions represented in the state model for the maintenance mode will be executed. The maintenance mode state functions are not necessarily the same functions as those in other modes, even though the unit states are named the same – they may be uniquely referenced for the mode they are associated with. Programmatically, the functions for each unit control mode are executed only when the respective unit control mode has been chosen by the “mode management” routine.

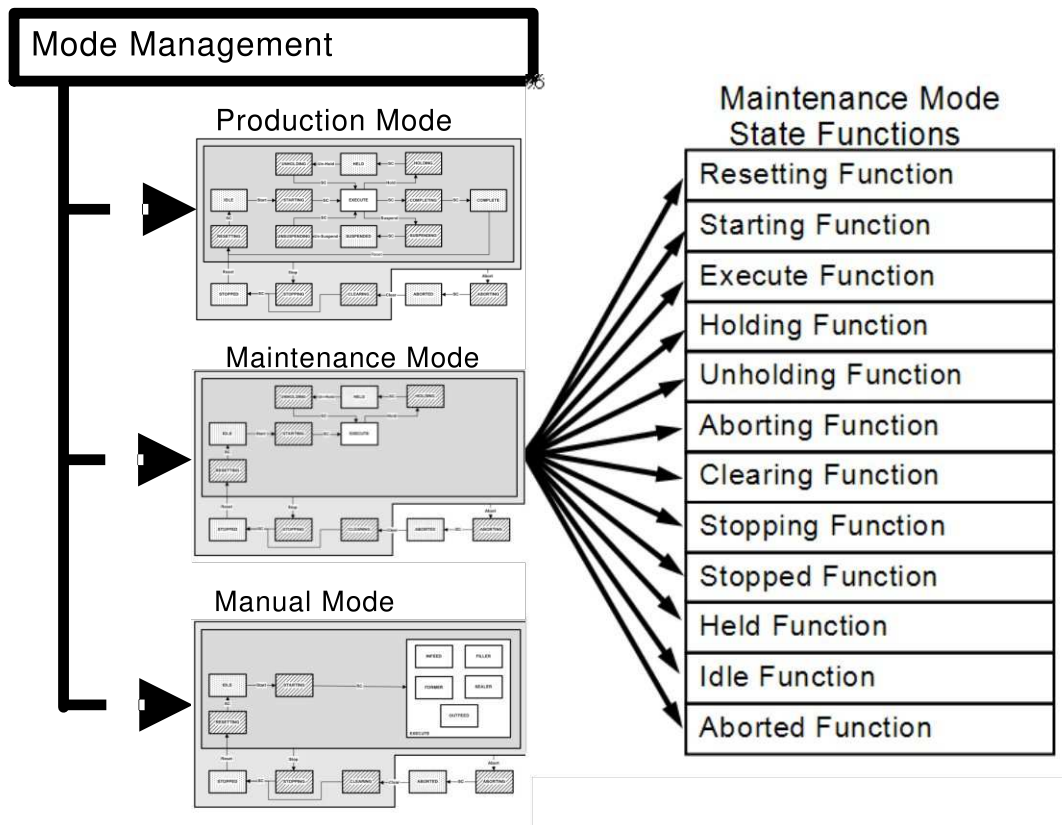


Figure 7: Maintenance mode execution model

6.3 Manual mode

Manual mode provides suitably authorized personnel the ability to operate individual subordinate equipment controls (such as drive logic) within the machine under manual pushbutton control. Such controls in this mode may be on a "hold-to-run" basis such that removal of the run signal will cause the drive to be stopped. The ability to perform specific functions will be dependent upon mechanical constraints and interlocks. Manual mode will be of particular use for setting up the machine to work.

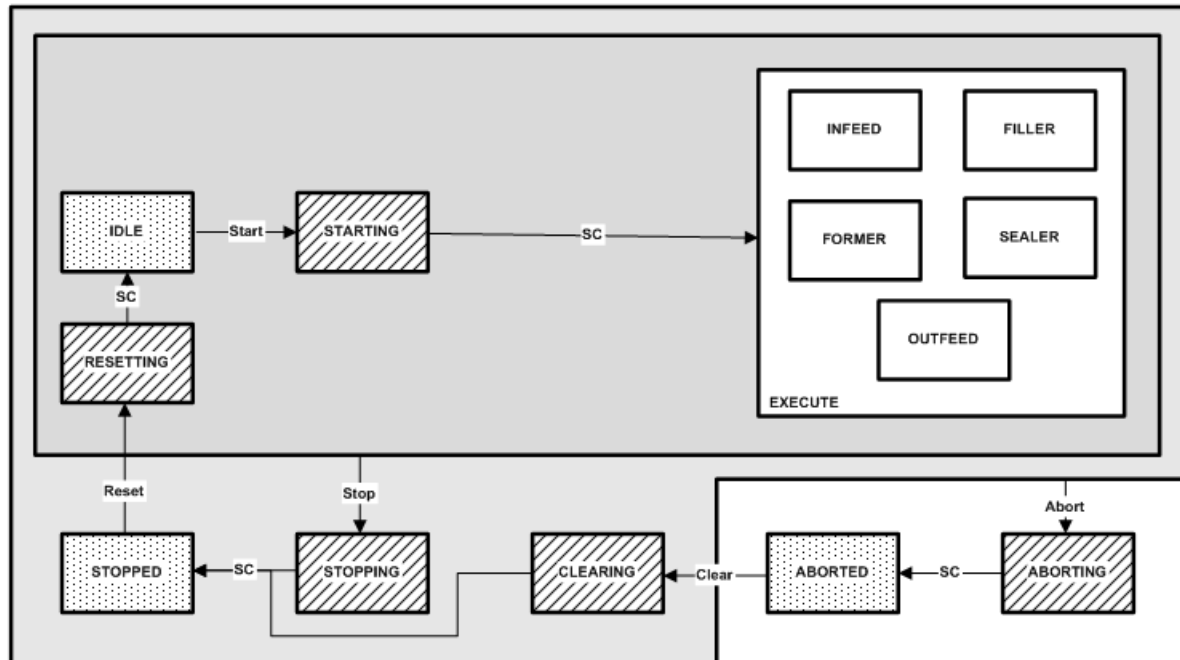


Figure 8: Manual mode state model

The predefined unit state model associated with manual mode can again be defined as a subset from the base state model. Common synonyms for this unit control mode are inch, jog, or index. Figure 8 illustrates a manual mode in which the EXECUTE states of subordinate equipment controls (such as drive logic) are shown within the EXECUTE state.

In Figure 9, the manual mode execution model is shown. Only unit state functions represented in the state model for the manual mode will be executed. The manual mode state functions are not necessarily the same functions as those in other modes, even though the unit states are named the same – they may be uniquely referenced for the mode they are associated with. Programmatically, the functions for each mode are executed only when the respective unit control mode has been chosen by the “mode management” routine.

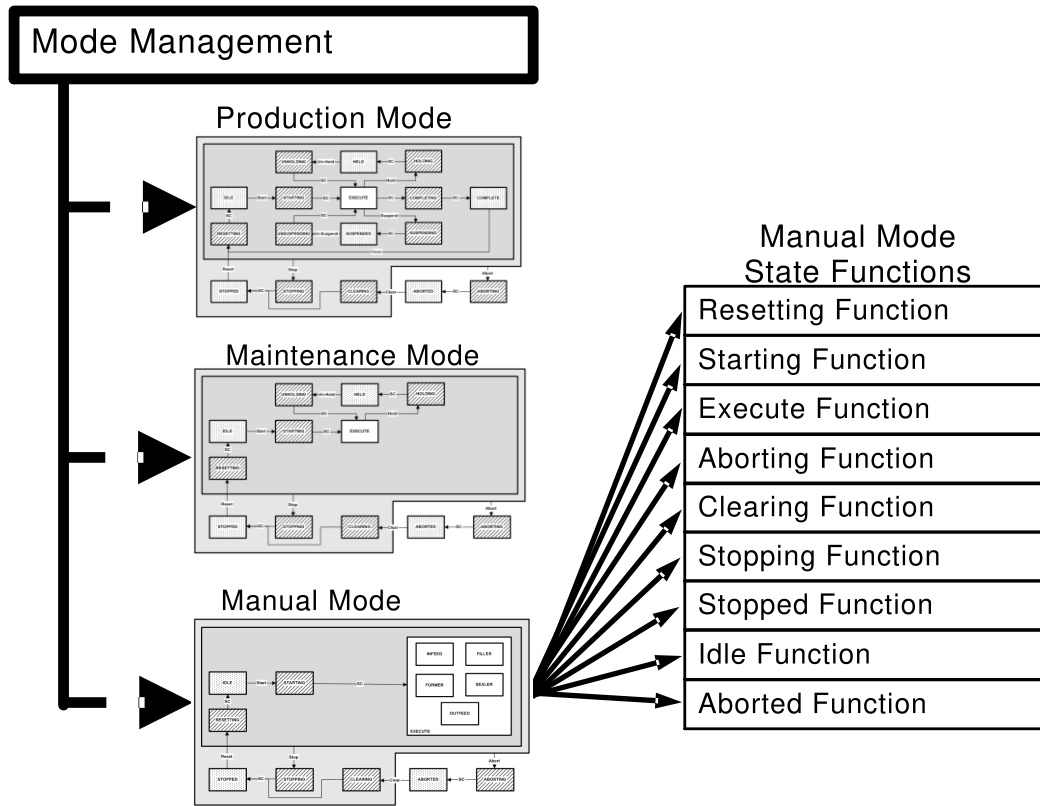
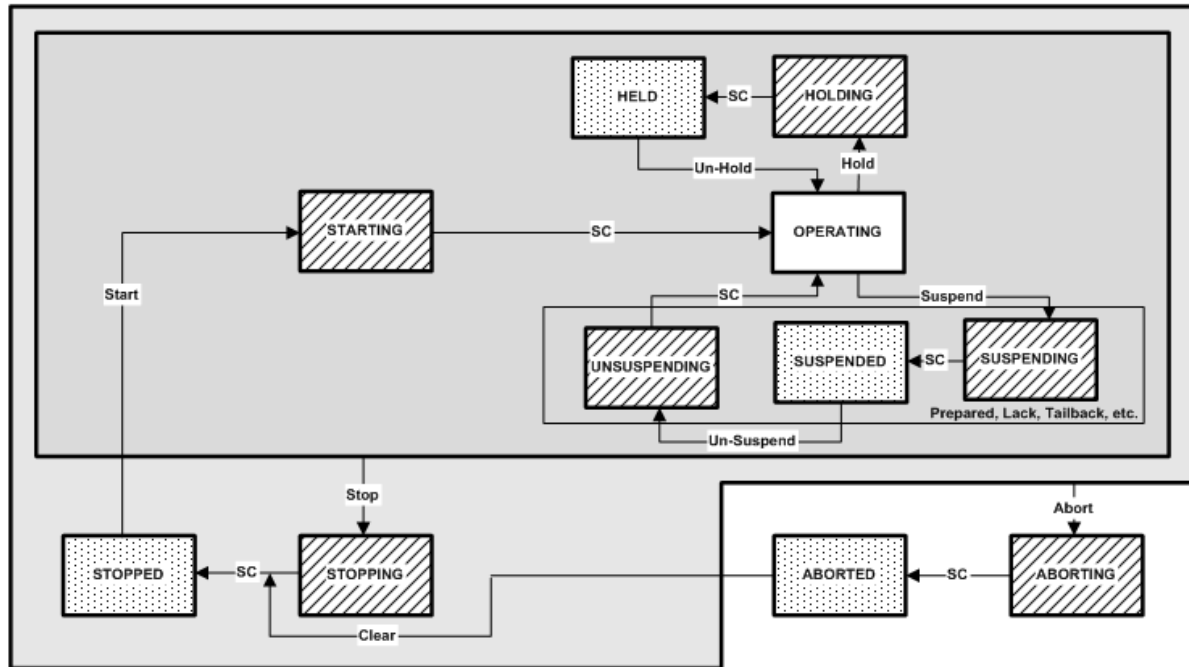


Figure 9: Manual mode execution model

6.4 User definable modes

Any unit control mode can be defined which provides a required function for the machine. The unit control mode provides for suitably authorized personnel operating the machine under pushbutton control, or for a remote system operating the machine as part of an integrated work center. This report recommends the approach in which all unit control modes are based on a fixed set of enumerated machine states. The “name” of the state(s) may be customized to provide the operator with an intuitive or descriptive name for the state(s), but the function of the state(s) is consistent with, and a subset of, the general definition of the base state model.

Below is a depiction of the Weihenstephan standard harmonized to the base state model in this report. In this user mode example the EXECUTE state is renamed the OPERATING state to be consistent with the terminology used in the Weihenstephan model. As can be seen, the base state model included states that were collapsed to be consistent with the Weihenstephan standard. The state commands are not defined as causing individual or undefined states; they are commands to implementation logic that describes the corresponding state transitions. There may be multiple conditions that cause a state transition but they do not cause “unique” machine states, they cause implementation specifics within the given framework of given states.



Note: OPERATING state is equivalent to an EXECUTE state; “prepared”, “lack”, and “tailback” conditions provide for the machine to go into the SUSPENDING state by activating the suspend command.

Figure 10: Automatic Weihenstephan state model