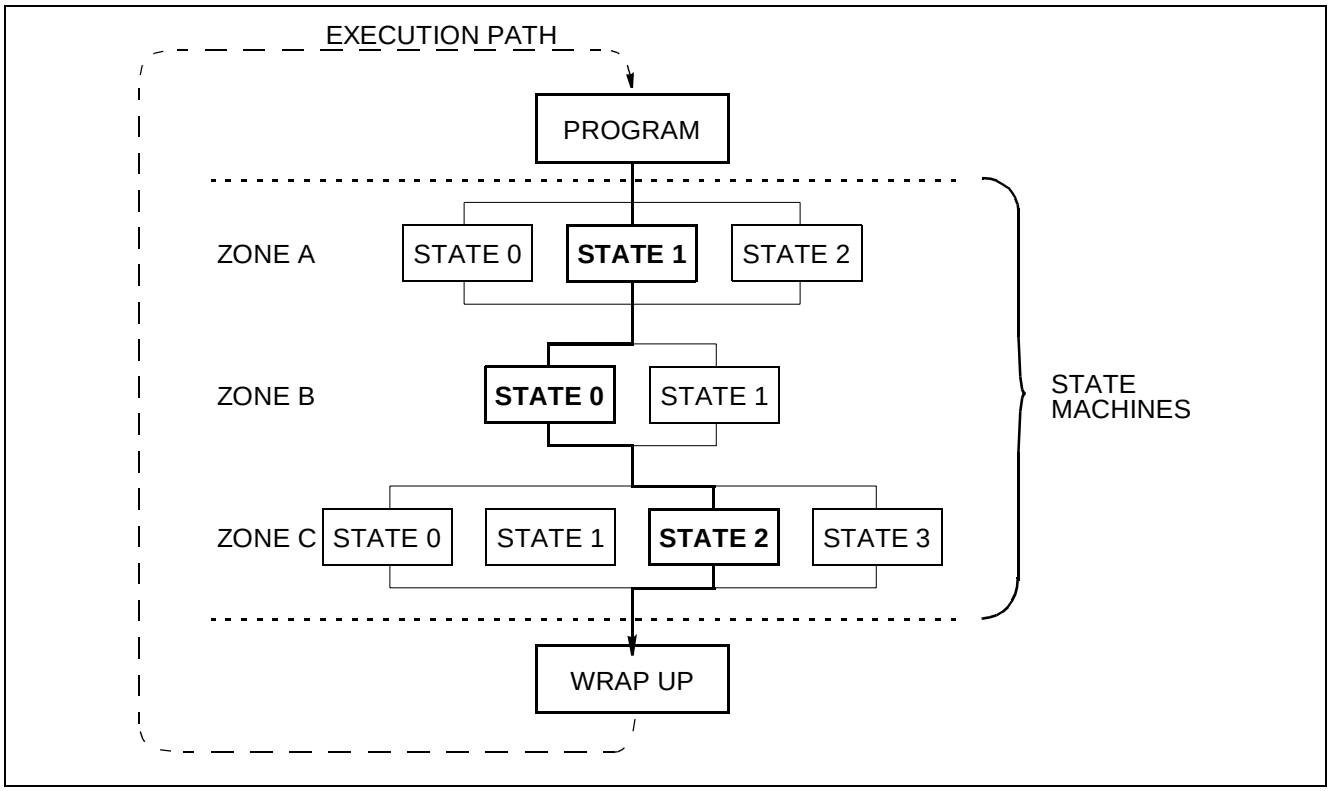
I/A Series® Remote Terminal Unit (RTU)
C50 State and Logic Language (SALL)



SALL provides traditional Boolean, sequential, and arithmetic calculations, as well as a High Level Serial Communications interface to IEDs, and 'State Machine' logic control. The ability of SALL to access all inputs and control all outputs, allows you to customize the RTU.

FEATURES

- C-like programming structure and syntax
- Access to all C50 input and output points
- Optional PID and AGA gas flow calculations
- User defined execution frequency, or event driven
- Debugging facilities available in TOOL diagnostic program
- SALL High Level Serial Interface (HLSI) supports serial interface to Intelligent Electronic Devices (IEDs)
- Integral to the operation of the C50 Configurator Program

SALL Overview

SALL files can be written using the text editing software of your choice. The source code is passed through the SALL pre-processor to produce C-language source, which is then compiled, linked and downloaded to the C50.

Floating point arithmetic is supported, making SALL ideal for running gas flow correction software (AGA), before transmitting data to the Master Station. PIDs can also be implemented, allowing closed loop control from the C50.

SALL provides a unique method of defining and utilizing serial data protocols in the C50, without making changes to the embedded firmware.

State and Logic Language

SALL allows the user to generate program modules designed from a sequential or state diagram. These modules are stored in the FLASH™ memory.

SALL supports and constructs expressions of the following types:

- Boolean, Integer, Timer and Floating Point
- Arithmetic expressions (+, -, *, /, <<, >>)
- Logic operations (AND, OR, NOT, XOR, INV)
- if-else
- Functions
- Arrays

SALL also supports Function statements, Timer elements, and Logic elements.

The SALL subsystem can run up to three independent logic modules, which can be executed at user selectable periods and priorities, or according to the occurrence of nominated events. For example, a program involving PID calculations may run every 150 milliseconds, while another may run each time one of five inputs changes state.

Electric Power algorithms can optionally be provided for Circuit Breaker Auto Reclose, Transformer Tap Control, Capacitor Bank Control, and Load Shed Coordination.

Standard AGA flow calculations (AGA-3, AGA-7, and NX-19) can optionally be provided for gas applications. A PID software control algorithm is also available for process control applications.

High Level Serial Interface

SALL provides a unique method of defining and implementing Async serial data protocols in the C50, without changing the embedded firmware. This facility allows users to program the serial interface to a variety of IEDs.

Serial protocol definition is done via a configuration utility program. This definition includes serial point setup, protocol framing information, and check value information. Check value information is comprised of CRC, checksum or user-defined check algorithm and data position within frame.

Once defined, this information is available to the SALL programmer, who uses embedded C50 functions to formulate messages to external devices and decode incoming messages. Information can be passed between serial port messages and to the C50's internal database.

Timers

Standard timing functions allow programs to change their actions at set times. Delays can be inserted, or variables can be checked continually over a set period. Access to the RTU calendar clock allows function execution according to date and time.

Sections

SALL programs are comprised of three sections: *Program*, *State Machine,* and *Wrap Up*. Both the Program and Wrap Up sections are sequential in their execution.

The 'State Machine' section has a number of zones. Each zone executes the code associated with only one state — a structure that simplifies the engineering of sound application logic.

Non-Volatile Variables

Non-volatile memory allows the C50 to retain variable values when power is lost, or when the RTU is reset.

SALL code can store variables in non-volatile memory if required allowing critical variables to be easily modified dynamically, but still be available after a reset, or power loss up to eight hours.

Debugging

All accesses to/from SALL use pre-validated mechanisms that make it virtually impossible for a SALL program to crash the RTU.

To assist in finding problems:

- The diagnostic program *TOOL* allows for the viewing of all variables
- The program can be stopped and run one pass at a time

By forcing the value of one variable to another value, and stopping the program after each pass, you can locate errors.

These features allow the testing of programs while simulating events.

FUNCTIONAL SPECIFICATIONS

Modules
IBM PC/AT compatible
Turbo C++ V3 for DOS

Maximum Number of Programs
Three

Period
0 (event driven) - 32767 milliseconds

Supported Data Types
I/O TYPES
Analog input, Digital input, Non-volatile Digital, SOE, Control (Latch, Trip/Close, Setpoint, Raise/Lower), Floating Point Analog
INTERNAL VARIABLE TYPES
Boolean, Integer, Floating Point, Timer, Zone Variable, Array

Maximum Size of SALL Executables
Program 1 - 24 Kb
Program 2 - 24 Kb
Program 3 - 16 Kb

Flow Control
if, if-else, state machines

Assignment Operator
=

Shift Operators
>>, <<, (SHR, SHL)

Logical Operators
&&, ||, (AND, OR)

Relational Operator
s <=, >=, <, >, = =, !=, (LE, GE, LT, GT, EQ, NE)

Bitwise Operators
&, ^, | (BIT_AND, BIT_XOR, BIT_OR)

Unary Operators
!, ~, - (NOT, COMP, NEG)

Binary Operators
-, +, *, /, MOD

Accumulator Functions
Add_to_Dicnt, Dicnt, Freeze, Freeze and Reset

Output Functions
Output, OutputQ, OutputQueuedControls

Timing Functions
Every, TimeDelayOn, TimeDelayOff, Pulse, RetriggablePulse, TimeLeft

Transition Functions
LL, HL, Low-to-High, High-to-Low

Floating Point Function
Int_to_FP, FP_to_Int, FP_Comp, FP_Add, FP_Sub, FP_Mul, FP_Div, FP_Sin, FP_Cos, FP_Exp, FP_Ln, FP_Pow, FP_Sqrt

Industry Functions
AGA3, AGA7, AGA8, NX19

Serial Interface Functions
Serial_Interface, Serial_Write, Serial_Read, Output_Packet, Get_Packet and twelve data conversion functions

The Foxboro Company
33 Commercial Street
Foxboro, Massachusetts 02035-2099
United States of America
http://www.foxboro.com
Inside U.S.: 1-508-543-8750 or 1-888-FOXBORO (1-888-369-2676)
Outside U.S.: Contact your local Foxboro representative.

MB 021 Printed in U.S.A. 0697

An Invensys company