plc - an experimental extension library for "plumbing"

Steve Pretty 07MAR11

"plc" is a library of software written in occam-pi, intended to be used in conjunction with the plumbing library (see www.concurrency.cc) on the Arduino family of microcontroller boards.

"plc" aims to explore ideas for extending the functionality of the existing plumbing library. The inspiration for "plc" has been drawn from the world of programmable logic controllers as described by IEC 61131 – particularly the functional block programming model. (Note – the author has no practical experience using industrial PLCs – his primary source of reference has been the text "Programmable Logic Controllers, 5th Edition" by W. Bolton, Newnes 2009).

"plc" has been divided into 6 modules, which are outlined below. Each module included a number of functional blocks. The API for each functional block is fully described in the code. There is at least one test case provided for each functional block, which provides additional documentation of the intended use for each of the blocks.

"plc" is a work in progress – more modules and functional blocks will be added over time. PLCs handle analogue signals as well as digital ones – so that area is next on the roadmap.

Key Design Decisions

- The plc library builds on "plumbing" – and "plumbing.module must be included at the head of any code using this library.
- The plc library uses the LEVEL type defined in plumbing to represent binary signals
- The INT type will be used to represent analogue signals.
- The plc library is implemented as a set of occam-pi PROCs
- The aim is to allow users to compose complex system behaviour by simply declaring the necessary functional blocks under a PAR statement, and linking the functional blocks using a set of CHANs. The user is of course free to use the library in conjunction with more complex occam-pi code.

plcBasic.module

- plc.NOT a logic inversion function
- plc.LD digital input. Pins 2 and 3 are interrupt driven, other pins are timed poll
- plc.LDN inverted digital input – to simulate normally closed switch contacts
- plc.OUT digital output
- plc.IR2 "internal relay" – multicasts a LEVEL message to two destinations
- plc.IR3 as above – three destinations
- plc.IR4 as above – four destinations
- plc.BH a black hole function – receives and discards unwanted LEVEL messages

plcLogic.module

- plc.OR, plc.OR3, plc.OR4 2, 3 and 4 input logic OR gates
- plc.AND, plc.AND3, plc.AND4 2, 3 and 4 input logic AND gates
- plc.NOR, plc.NOR3, plc.NOR4 2, 3 and 4 input logic NOR gates
- plc.NAND, plc.NAND3, plc.NAND4 2, 3 and 4 input logic NAND gates
- plc.XOR 2 input logic Exclusive OR gate
- plc.EQ 2 input logic Equals gate
- plc.MUX 2 input multiplexer

plcLatches.module

- plc.SR Asynchronous Set / Reset gate
- plc.FF Asynchronous Flip Flop (toggle) gate
- plc.D Synchronous D type flip flop
- plc.JK Synchronous JK flip flop

plcTiming.module

- plc.TP A single pulse timer (retriggerable monostable) – simple version
- plc.TPX Extended functionality version of plc.TP
- plc.TON Turn on timer – output goes high fixed period after input goes high
- plc.TOF Turn off timer – output stays high for fixed period after input goes low
- plc.CLK Gateable clock generator.

plcCounters.module

- plc.CYCNT Simple cyclic count up counter
- plc.CTU Extended functionality count up counter

plcSequencers.module

- plc.SEQ4 A 4 output logic sequencer / decoder
- plc.SEQ8 An 8 output logic sequencer / decoder