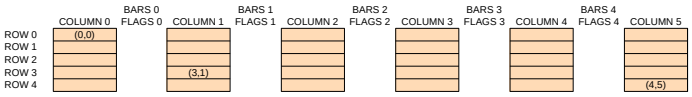
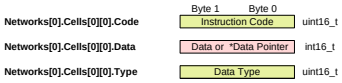
Network Structure

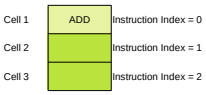


Cell Structure:



Memory allocation in Network structure for different instructions

For blocks (instructions) that uses more than one cell, "Instruction Index" has the index for this cell, example:
Used by Logic Evaluator and Logic Editor



		Byte 1	Byte 0		
TIMER	Networks[0].Cells[0][0].Code	Instruction Code		Uint 16	Instruction Code
	Networks[0].Cells[0][0].Data	*Data Pointer		Int 16	Pointer to T[x] array, actual value of Timer in ms (32 bits unsigned long)
	Networks[0].Cells[0][0].Type	Data Type		Uint 16	Always 10 for Timers
		Byte 1	Byte 0		
Second cell	Networks[0].Cells[0][0].Code	Instruction Code		Uint 16	Same Instruction Code + Index of instructions bigger than one cell
	Networks[0].Cells[0][0].Data	*Data Pointer		Int 16	Setpoint of Timer
	Networks[0].Cells[0][0].Type	Data Type		Uint 16	Scale for Timer. Direct division: 1 ms, 10 0.01 sec, 100 0.1 sec, 1000 sec, 60000 min

Byte 1	Byte 1	Byte 0	
Index	Instruction Code 12 bits		
0 0 0 0			Instruction is simple height cell or it is the first cell of a multiple cells instruction
x 0 0 1			Second cell for a multiple cell height instruction
x 0 1 0			Third cell for a multiple cell height instruction
x 0 1 1			Fourth cell for a multiple cell height instruction
x 1 0 0			Fifth cell for a multiple cell height instruction
1 x x x			Second column for a multiple columns instruction

COUNTER	Networks[0].Cells[0][0].Code	Instruction Code	Uint 16	Instruction Code
	Networks[0].Cells[0][0].Data	*Data Pointer	Int 16	Pointer to T[x] array, actual value of Timer in ms (32 bits unsigned long)
	Networks[0].Cells[0][0].Type	Data Type	Uint 16	Always 9 for Counters
Second cell	Networks[0].Cells[0][0].Code	Instruction Code	Uint 16	Same Instruction Code + Index of instructions bigger than one cell
	Networks[0].Cells[0][0].Data	*Data Pointer	Int 16	Setpoint of Counter
	Networks[0].Cells[0][0].Type	Data Type	Uint 16	Not used

SUB, ADD MUL, DIV	Networks[0].Cells[0][0].Code	Instruction Code	Uint 16	Instruction Code
	Networks[0].Cells[0][0].Data	*Data Pointer	Int 16	Pointer to First Data
	Networks[0].Cells[0][0].Type	Data Type	Int 16	Type for first data
Second cell	Networks[0].Cells[0][0].Code	Instruction Code	Uint 16	Same Instruction Code + Index of instructions bigger than one cell
	Networks[0].Cells[0][0].Data	*Data Pointer	Int 16	Pointer to Second Data
	Networks[0].Cells[0][0].Type	Data Type	Int 16	Type for second data
Third	Networks[0].Cells[0][0].Code	Instruction Code	Uint 16	Same Instruction Code + Index of instructions bigger than one cell
	Networks[0].Cells[0][0].Data	*Data Pointer	Int 16	Pointer to Third data
	Networks[0].Cells[0][0].Type	Data Type	Int 16	Type for third data

Instruction Codes

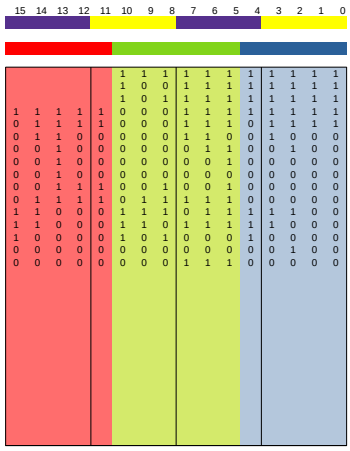
Code	Mnemonic	Canvas	Programmed	Group	Heigth	Width	Description
0	NOP				1	1	No instruction is present in this cell. Empty Cell
1	CON	drawConn	execConn	Basic	1	1	Direct connection
2	INV	drawNeg	execNeg	Basic	1	1	Direct connection – Inverted
3	NO	drawNO	execNO	Basic	1	1	Normal Open Contact
4	NC	drawNC	execNC	Basic	1	1	Normal Closed Contact
5	RE	drawRE	execRE	Basic	1	1	Rise Edge contact
6	FE	drawFE	execFE	Basic	1	1	Fall Edge Contact
7	C	drawCoil	execCoil	Basic	1	1	Coil
8	L	drawCoilL	execCoilL	Basic	1	1	Latch Coil (Set)
9	U	drawCoilU	execCoilU	Basic	1	1	Unlatch Coil (Reset)
10	TON	drawTon	execTON	Timers	2	1	Timer On
11	TOFF	drawToff	execTOFF	Timers	2	1	Timer Off
12	TP	drawTp	execTP	Timers	2	1	Timer Pulse
13	CTU	drawCTU	execCTU	Counters	2	1	Counter Up
14	CTD	drawCTD	execCTD	Counters	2	1	Counter Down
15	MOVE	drawMove	execMOVE	Basic Math	2	1	Move
16	SUB	drawSUB	execSUB	Basic Math	3	1	16bit int Subtraction
17	ADD	drawADD	execADD	Basic Math	3	1	16bit int Addition
18	MUL	drawMUL	execMUL	Basic Math	3	1	16bit int Multiplication
19	DIV	drawDIV	execDIV	Basic Math	3	1	16bit int Division
20	MOD	drawMOD	execMOD	Basic Math	3	1	16bit int Module of division (remainder)
21	SHL	drawSHL	execSHL	Binary (Bitwise)	2	1	Shift left
22	SHR	drawSHR	execSHR	Binary (Bitwise)	2	1	Shift right
23	ROL	drawROL	execROL	Binary (Bitwise)	2	1	Rotate Left
24	ROR	drawROR	execROR	Binary (Bitwise)	2	1	Rotate Right
25	AND	drawAND	execAND	Binary (Bitwise)	3	1	Bitwise And
26	OR	drawOR	execOR	Binary (Bitwise)	3	1	Bitwise Or
27	XOR	drawXOR	execXOR	Binary (Bitwise)	3	1	Bitwise Xor
28	NOT	drawNOT	execNOT	Binary (Bitwise)	2	1	Bitwise Not
29							
30							
31							
32							
33							
34							
35							
36							
37							
38							
39							
40							
41							
42							
43							
44							
45							
46							

Instructions to be implemented

SUB MUL, etc for floating point or Int (32 bits)		Visualization problem	
CONVERSION	general conversion blocks, specially between 16bit and real		
PID	only REAL ?	Control	
RAMP	only REAL ?	Control	
INTEGRATE	only REAL	Control	Integral calculation
DERIVATIVE	only REAL	Control	derivate the input
HYSTERESIS	only REAL	Control	bool out between 3 values real
ABS	only REAL	Advanced Math	
SQRT	only REAL	Advanced Math	
LN	only REAL	Advanced Math	
LOG	only REAL	Advanced Math	
EXP	only REAL	Advanced Math	
SIN	only REAL	Trigonometry	
COS	only REAL	Trigonometry	
TAN	only REAL	Trigonometry	
ASIN	only REAL	Trigonometry	
ACOS	only REAL	Trigonometry	
ATAN	only REAL	Trigonometry	
RAN	only REAL	Random value generator	
FIFO			
LIFO			
RBUF	Ring Buffer		
TIME and DATE	schedule management		
	arithmetic on Time and Date		
String Functions			
SEL	not really needed	Selection	Select between 2 inputs
MAX	not really needed	Selection	Max between 2 inputs
MIN	not really needed	Selection	Min between two inputs
LIM	not really needed	Selection	Limit for a given input between two threshold
MUX	not really needed	Selection	Select between 2 inputs or more
GT	not really needed	Compare	Greater than
GE	not really needed	Compare	Greater than or Equal to
EQ	not really needed	Compare	Equal to
LT	not really needed	Compare	Less than
LE	not really needed	Compare	Less than or equal to
NE	not really needed	Compare	Not equal to

Types

Tag	Type	DataType Code	Quantity	Example	Description	Comment
M	Byte	0	2000	M0.0	Internal Marks or Flags	
Q	Byte	1	200	Q0.0	Digital Outputs	
I	Byte	2	200	I0.0	Digital Inputs	
Cd	Byte	3	200	Cd1 Cd199	Counter Activation bit	Image duplicated to get the Edges with easy user programming, no extra bit required
Cr	Byte	4	200	Cr1 Cr199	Counter Running bit	
Td	Byte	5	300	Td1 Td200	Timer Activation bit	
Tr	Byte	6	300	Tr1 Tr200	Timer Running bit	
IW	uint16_t	7	50	IW0 IW56	Analog Inputs	
QW	uint16_t	8	50	QW1 QW17	Analog Outputs	
C	uint16_t	9	200	C1 C154	Counter Register	
T	struct	10	200	T1 T34	Timer Register	
D	int16_t	11	10000	D500 D1	Regular Integer Registers	
K	-	12	na	K0 K-30888	Integer constant value	
R	Real	13	2000	R1 R1523	Floating Points Registers	Not yet implemented
KR	-	14	na	KR(min real) KR(max real)	Real Constant value	Not yet implemented



Custom colors
cyan
aqua
cyanl
pink
purple
purple1
purple2
brown
brown1
yellow1
yellow2
White1
White2
White3
darkblue
LightGreen

	Display Connector / SPI
	USB / TX-RX
	IO available
	Flash Control
	IO Expansion
	SD card CS

PLsi v0	WROOM32 DevKitC	Index	WROOM32 DevKitC	PLsi v0
	3V3	1	GND	
	EN	2	GPIO23 (Display)	
In7 / analog in 1	GPIO36 (ADC1 CH0)	3	GPIO22	I2C SCL
In6 / analog in 0	GPIO39	4	GPIO1 TX	
in0	GPIO34	5	GPIO3 RX	
in1	GPIO35	6	GPIO21	I2C SDA
in2	GPIO32	7	GND	
	GPIO33 (Display)	8	GPIO19 (Display)	
out5 / analog out 1	GPIO25	9	GPIO18 (Display)	
out4 / analog out 0	GPIO26	10	GPIO5 (SD CS)	SD_CS
	GPIO27 (Display)	11	GPIO17	out0
	GPIO14 (Display)	12	GPIO16	out1
	GPIO12 (Display)	13	GPIO4	in4
	GND	14	GPIO0	in5
in3	GPIO13	15	GPIO2	out2
	SD2	16	GPIO15	out3
	SD3	17	SD1	
	CMD	18	SD0	
	5V	19	CLK