Softlog Systems

Softlog Systems (2006) Ltd.

ICP Family Programmers

User's Manual

IMPORTANT NOTE:

- Starting from year 2019 Softlog Systems manufactures ICP2(**G3**), ICP2-GANG(**G3**), ICP2-COMBO(**G3**) and ICP2-Portable(**G3**) programmers instead of legacy ICP2, ICP2-GANG, ICP2-COMBO and ICP2-Portable
- Due to nearly full compatibility all of them are referred below as ICP2, ICP2-GANG, ICP2-COMBO and ICP2-Portable respectively. If difference is applied then they are referred as "G3 products" and "non-G3 products"

1 Connectors

1.1 "Power" Connector (Power Jack, Center Pin 2.1mm)

Pin No.	*Pin Name: ICP2-GANG, ICP2-COMBO*	*Pin Name (ICP2, ICP2(HC), ICP2-Portable)*	*Voltage Range*
1/center	POWER (+)	POWER (+/-)	12V to 15V 
2	POWER (-)	POWER (+/-)	

1.2 "USB" Connector

- ICP2-Portable(G3) only: Mini-USB
- Other programmers: USB Type-B

Note: galvanically isolated on ICP2-COMBO

1.3 "LAN" Connector: Standard RJ-45

Note: available on ICP2-COMBO only

1.4 "RS-232 IN" Connector (Standard D-type 9 Female)

Notes:
- not available on ICP2-Portable
- galvanically isolated on ICP2-COMBO

Pin No.	*Pin Name*	*Voltage Range*	*Pin Type*	*Description*
1	-	-	-	Not connected
2	**PC_RXD**	RS-232 level	RS-232 output	TxD output to PC
3	**PC_TXD**	RS-232 level	RS-232 input	RxD input from PC
4	PC_DTR	+12 to +15V	Power	**ICP2-GANG, ICP2-COMBO**: not connected **ICP2/ICP2(HC)**: additional power supply input
5	**GND**	-	GND	Ground connection
6	12V_OUT	+11…14V	Power	**ICP2-GANG**: power output **ICP2/ICP2(HC)/ICP2-COMBO**: not connected
7,8,9	-	-	-	Not connected

1.5 "RS-232 OUT" Connector (Standard D-type 9 Male)

Note: available on ICP2-GANG only

Pin No.	*Pin Name*	*Voltage Range*	*Pin Type*	*Description*
1	-	-	-	Not connected
2	CHAIN_232_RXD	RS-232 level	RS-232 input	RxD input from next ICP2-GANG
3	CHAIN_232_TXD	RS-232 level	RS-232 output	TxD output to next ICP2-GANG
4	-	-	-	Not connected
5	GND	-	GND	Ground connection
6,7,8,9	-	-	-	Not connected

1.6 "Control Interface" Connector (DIN-64 A, B Male)

Notes:
- available on ICP2-COMBO only
- all voltages are referenced to isolated GND (GND_ISO)

Mating connector: 09022646421 (Harting) - DIN41612, 2 rows, 64 pins, female, B style, receptacle, straight

Pin No.	*Pin Name*	*Pin Type*	*Description*
A1	-		
B1	5V_ISO	5V reference source	Isolated 5V with in-series fuse 50mA, can be used to control ENV_SEL_A/B/C_ISO lines
A2	GND_ISO	-	Isolated GND
B2	GND_ISO	-	Isolated GND
A3	PC_RXD_ISO	Standard RS-232 output	RS-232 output to PC
B3	PC_TXD_ISO	Standard RS-232 input	RS-232 input from PC
A4	CHAIN_232_RXD_ISO	Standard RS-232 input	RS-232 input from chained ICP2-COMBO(G3)
B4	CHAIN_232_TXD_ISO	Standard RS-232 output	RS-232 output to chained ICP2-COMBO(G3)
A5	GND_ISO	-	Isolated GND
B5	GND_ISO	-	Isolated GND
A6	ENV_SEL_A_ISO	CMOS input (0/5V) with weak pull-down	Environment selection: *…C… …B… …A… Result* 0 0 0 Env. 1 0 0 1 Env. 2 0 1 0 Env. 3 0 1 1 Env. 4 1 0 0 Env. 5 1 0 1 Env. 6 1 1 0 Reserved 1 1 1 Reserved Reserved combination selects Env. 1
B6	ENV_SEL_B_ISO	CMOS input (0/5V) with weak pull-down	
A7	ENV_SEL_C_ISO	CMOS input (0/5V) with weak pull-down	
B7	-	-	-
A8	-	-	-
B8	GND_ISO	-	Isolated GND
A9	GO_ISO_1	CMOS input (0/Z) with weak pull-up	Ch. 1: programming start (active low)
B9	GO_ISO_2	CMOS input (0/Z) with weak pull-up	Ch. 2: programming start (active low)
A10	GO_ISO_3	CMOS input (0/Z) with weak pull-up	Ch. 3: programming start (active low)
B10	GO_ISO_4	CMOS input (0/Z) with weak pull-up	Ch. 4: programming start (active low)
A11	GO_ISO_5	CMOS input (0/Z) with weak pull-up	Ch. 5: programming start (active low)
B11	GO_ISO_6	CMOS input (0/Z) with weak pull-up	Ch. 6: programming start (active low)
A12	GO_ISO_7	CMOS input (0/Z) with weak pull-up	Ch. 7: programming start (active low)
B12	GO_ISO_8	CMOS input (0/Z) with weak pull-up	Ch. 8: programming start (active low)
A13	GO_ISO_9	CMOS input (0/Z) with weak pull-up	Ch. 9: programming start (active low)
B13	GO_ISO_10	CMOS input (0/Z) with weak pull-up	Ch. 10: programming start (active low)
A14	GO_ISO_11	CMOS input (0/Z) with weak pull-up	Ch. 11: programming start (active low)
B14	GO_ISO_12	CMOS input (0/Z) with weak pull-up	Ch. 12: programming start (active low)
A15	-	-	-
B15	-	-	-
A16	-	-	-
B16	-	-	-
A17	PASS_ISO_1	CMOS output with in-series R=330Ω	Ch. 1: pass/busy indication
B17	PASS_ISO_2	CMOS output with in-series R=330Ω	Ch. 2: pass/busy indication
A18	PASS_ISO_3	CMOS output with in-series R=330Ω	Ch. 3: pass/busy indication
B18	PASS_ISO_4	CMOS output with in-series R=330Ω	Ch. 4: pass/busy indication
A19	PASS_ISO_5	CMOS output with in-series R=330Ω	Ch. 5: pass/busy indication
B19	PASS_ISO_6	CMOS output with in-series R=330Ω	Ch. 6: pass/busy indication
A20	PASS_ISO_7	CMOS output with in-series R=330Ω	Ch. 7: pass/busy indication
B20	PASS_ISO_8	CMOS output with in-series R=330Ω	Ch. 8: pass/busy indication
A21	PASS_ISO_9	CMOS output with in-series R=330Ω	Ch. 9: pass/busy indication
B21	PASS_ISO_10	CMOS output with in-series R=330Ω	Ch. 10: pass/busy indication
A22	PASS_ISO_11	CMOS output with in-series R=330Ω	Ch. 11: pass/busy indication
B22	PASS_ISO_12	CMOS output with in-series R=330Ω	Ch. 12: pass/busy indication
A23	-	-	-
B23	-	-	-
A24	-	-	-
B24	-	-	-

Pin No.	Pin Name	Pin Type	Description
A25	FAIL_ISO_1	CMOS output with in-series R=330Ω	Ch. 1: fail/busy indication
B25	FAIL_ISO_2	CMOS output with in-series R=330Ω	Ch. 2: fail/busy indication
A26	FAIL_ISO_3	CMOS output with in-series R=330Ω	Ch. 3: fail/busy indication
B26	FAIL_ISO_4	CMOS output with in-series R=330Ω	Ch. 4: fail/busy indication
A27	FAIL_ISO_5	CMOS output with in-series R=330Ω	Ch. 5: fail/busy indication
B27	FAIL_ISO_6	CMOS output with in-series R=330Ω	Ch. 6: fail/busy indication
A28	FAIL_ISO_7	CMOS output with in-series R=330Ω	Ch. 7: fail/busy indication
B28	FAIL_ISO_8	CMOS output with in-series R=330Ω	Ch. 8: fail/busy indication
A29	FAIL_ISO_9	CMOS output with in-series R=330Ω	Ch. 9: fail/busy indication
B29	FAIL_ISO_10	CMOS output with in-series R=330Ω	Ch. 10: fail/busy indication
A30	FAIL_ISO_11	CMOS output with in-series R=330Ω	Ch. 11: fail/busy indication
B30	FAIL_ISO_12	CMOS output with in-series R=330Ω	Ch. 12: fail/busy indication
A31	-	-	-
B31	-	-	-
A32	-	-	-
B32	-	-	-

1.7 "TARGET" Connector D-type 15 Female: all programmers excluding ICP2-COMBO

Notes: ICP2-GANG - 4 identical channels
ICP2/ICP2(HC)/ICP2-Portable - 1 channel

Pin No.	Pin Name	Voltage Range	Pin Type	Description
1	T_VDD	2.0V to 5.5V	Output or input with weak pull-down and programmable strong pull-down	Target VDD supply voltage
2	GND	-	-	Ground connection
3	T_SCK **(4)**	2.0V to 5.5V	CMOS output or input with weak pull-down	Target clock
4	T_MOSI **(4)**	2.0V to 5.5V	CMOS output or input with weak pull-down	Target data (out, in/out)
5	T_MISO **(4)**	2.0V to 5.5V	CMOS output or input with weak pull-down	Target data (in)
6	T_VPP/MCLR	2.0V to 13.5V	Output or input with weak pull-down	Target VPP supply voltage
7	T_TARG **(2)**	5.0V	CMOS output	General purpose output
8	T_VTEST (non-G3 products)	2.0V to 13.5V	Output	Target VTEST signal for PIC17Cxxx family
	T_DIO_2 (G3 products)	2.0V to 5.5V	Output or input with weak pull-down	Target I/O number 2
9	T_DIO_0	2.0V to 5.5V	CMOS output or input with weak pull-down	Target I/O number 0 Non-G3 products: FOSC signal for PIC17Cxxx family
10	T_DIO_1	2.0V to 5.5V	CMOS output or input with weak pull-down	Target I/O number 1
11	GND	-	-	Optional ground connection
12	GND **(1)**	-	-	Optional ground connection
13	GO **(1,3)**	0-1.0V or N/C	CMOS input with pull-up 10K	Input for programming start in standalone mode
14	PASS_OUT**(1,3)**	5.0V	CMOS output	Output for pass/fail/busy indication
15	FAIL_OUT **(1,3)**	5.0V	CMOS output	Output for pass/fail/busy indication

Notes:
(1) Dedicated for standalone operation without PC
(2) Not available on ICP2-Portable (non-G3)
(3) Optional for ICP2-Portable, contact Softlog Systems for details
(4) G3 products additionally contain strong (2.2K) programmable pull-up and pull-down resistors

1.8 ICP2-COMBO only: "TARGET" Connector (DIN-48, A,B,C, male); 1, 2 or 3 Identical Connectors

Mating connector: 86093488314755V1LF (FCI) - DIN41612, 3 rows, 48 pins, female, C/2 style, straight

Pin Number				***Pin Name***	***Opto-relay barrier***	***Pin Type***	***Description***
CH. 1 CH. 5 CH. 9	***CH. 2 CH. 6 CH. 10***	***CH. 3 CH. 7 CH. 11***	***CH. 4 CH. 8 CH. 12***				
A1	A5	A9	A13	T_VPP/ MCLR	Yes	Power output or input with weak pull-down	Target VPP/MCLR supply voltage
A2	A6	A10	A14	T_SCK **(4)**	Yes	CMOS output or input with weak pull-down	Target clock
A3	A7	A11	A15	T_MOSI **(4)**	Yes	CMOS output or input with weak pull-down	Target data (out, in/out)
A4	A8	A12	A16	T_DIO_0	Yes	CMOS output or input with weak pull-down	Target I/O number 0 Non-G3 products: FOSC signal for PIC17Cxxx family
B1	B5	B9	B13	T_DIO_1	Yes	CMOS output or input with weak pull-down	Target I/O number 1
B2	B6	B10	B14	T_MISO **(4)**	Yes	CMOS output or input with weak pull-down	Target data (in)
B3	B7	B11	B15	T_VTEST (non-G3 products)	Yes	Output	Target VTEST signal for PIC17Cxxx family
				T_DIO_2 (G3 products)		CMOS output or input with weak pull-down	Target I/O number 2
B4	B8	B12	B16	T_VDD	Yes	Power output or input with weak pull-down and programmable strong pull-down	Target VDD supply voltage
C1	C5	C9	C13	GND_SW	Yes	-	Ground connection via opto-relay barrier. **Not recommended**
C2, C3	C6, C7	C10, C11	C14, C15	GND	-	-	Ground connection (permanent)
C4	C8	C12	C16	T_TARG	-	CMOS output	General purpose output

Notes:
(4) G3 products additionally contain strong (2.2K) programmable pull-up and pull-down resistors

1.9 Typical Connection to "TARGET" Connector

- See separate document "ICP2(G3) Connection Table.pdf" for the latest table

Pin Name	*ICSP: PIC10/12/16, PIC18, PIC24, dsPIC,PIC32*	*I2C*	*Keeloq®*	*SPI Flash*	*Atmel SPI: ATmega, ATtiny*	*SWD*	*JTAG*	*UPDI (ATtiny, ATmega, AVR)*	*TPI (ATtiny)*	*PDI (ATxmega)*	*LPC80x, LPC17xx (UART)*
T_VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND
T_SCK	CLOCK (PGC)	SCL	CLOCK	SCK	SCK	SWCLK	TCK	-	TPICLK	PDI_CLK	-
T_MOSI	DATA (PGD)	SDA	DATA	SI	MOSI	SWDIO	TDI	UPDI	TPIDATA	PDI_DATA	UART0_RX
T_MISO	-	-	-	SO	MISO	-	TDO	-	-	-	UART0_TX
T_VPP	MCLR/VPP	-	-	-	RESET	RESET (12)	RESET(2)	-	RESET	-	RESET
T_TARG	-	-	-	-	-	-	-	-	-	-	-
T_DIO_2	-	-	-	CE	-	ERASE (1) ISP entry (3)	TEST (11)	-	-	-	ISP entry (4)
T_DIO_0	-	-	-	-	-	-	-	-	-	-	-
T_DIO_1	-	-	S1	-	-	-	TMS	-	-	-	-

Pin Name	*Microwire EEPROM (93C46, ...)*	*SWIM (STM8)*	*SWI (ATSHA204, ATECC608)*	*BDM (MC9S08)*	*CYBLE-013025 (UART)*	*C2 (EFM8)*	*SBW (MSP430)*				
T_VDD	VDD	VDD	VCC	VDD	VDD	VDD or VREGIN	VDD				
GND	GND	GND	GND	GND	GND	GND	GND				
T_SCK	CLK	-	-	-	-	C2CK	SBWTCK				
T_MOSI	DI	SWIM (5)	SDA (10)	BKGD (7)	UP_RX (9)	C2D	SBWTDIO				
T_MISO	DO	-	-	-	UP_TX	-	-				
T_VPP	-	RESET	-	- (8)	XRES (9)	-	-				
T_TARG	-	-	-	-	-	-	-				
T_DIO_2	CS	-	-	-	SDA (9)	-	-				
T_DIO_0	-	-	-	-	-	-	-				
T_DIO_1	ORG (6)	-	-	-	-	-	-				

Note 1: Atmel Cortex M7/M4 only (SAM E/S/V, SAM4, SAMG5x)
Note 2: RESET pin is not required for ATmega/ATxmega with JTAG interface
Note 3: ISP entry for LPC80x (recommended)
Note 4: PIO0_12 for LPC80x, P2.10 for LPC17xx – see NXP User's Manuals for details
Note 5: Resistor 1K±10% between SWIM and VDD is required
Note 6: ICP2(G3) family programmers set this pin according to selected memory organization: x8 = GND, x16 = VDD
Note 7: Resistor 2K-10K between BKGD and VDD is recommended
Note 8: No RESET pin connection is required but the RESET pin should have pull-up resistor
Note 9:
- Resistor 2.2K-10K between T_MOSI (D-15 pin 4) and GND (D-15 pin 2) must be connected in order to enter programming mode
- XRES connection with pullup about 10K is recommended for reliable programming
- SDA connection is recommended for possible recovery
Note 10: Resistor 1K between T_MOSI and VDD is required
Note 11: MSP430 only
Note 12: RESET pin is not used for nRF51xxx/nRF528xx devices

2 PC-Driven and Standalone Modes

ICP family programmers can be operated in PC-driven and/or standalone mode

Programmer	*PC-Driven*	*Standalone*
ICP2/ICP2(HC)	Yes	Yes
ICP2-GANG	Yes (**single** channel only)	Yes
ICP2-COMBO	Yes (**single** channel only)	Yes
ICP2-Portable	Yes	Yes

PC-driven mode means that all programming parameters and data are set in **PC** and the PC executes required sequences (programming, verification, blank check, etc.)

Standalone mode means that all programming parameters and HEX file data ("Environment", "PJ2 file") are saved in **programmer's** non-volatile flash memory. See paragraph 14 "Preparing Environment and Transferring Environment to Programmer".

Standalone programming can be activated by 2 ways:
- from PC
- by GO input on the programmer unit (NOTE: optional on ICP2-Portable)
True parallel multi-channel programming can be done in standalone mode only

3 Chain Connection (ICP2-GANG and ICP2-COMBO)

Number of channels can be increased (up to 64 channels) by daisy chain connection between ICP2-GANG or ICP2-COMBO units.

ICP2-GANG: see "ICP2-GANG Quick Start" manual for details
ICP2-COMBO: see "ICP2-COMBO-12 - Assign Address to Programmer" and "ICP2-COMBO-8 - Assign Address to Programmer" for details

IMPORTANT:
- all programmer units should have the **same** (all=yes or all=no) "DLL/Command Line Support" option
- all programmer units should be of the same generation: G3 or non-G3
- using short RS-232 cables is recommended, 3 wires only (TxD, RxD and GND)

4 PASS/FAIL LEDs and Outputs

##	*Conditions*	*PASS LED*	*FAIL LED*	*PASS Output*	*FAIL Output*
1.	Power-up	2 sec ON		2 sec ON	
2.	Operation in-progress (busy)	ON		ON	
3.	Programming done: **PASS**	ON	OFF	ON	OFF
4.	Programming done: **FAIL** (verification error)	OFF	ON	OFF	ON
5.	UUT problem during operation: - Vdd overload - Vpp overload - I2C communication error	OFF	Blink	OFF	ON
6.	Non-UUT problem during standalone operation: - database error - device not supported - no family activation - etc.	OFF	Blink	OFF	OFF
7.	Operation is canceled	OFF	OFF	OFF	OFF
8.	No firmware presents (bootloader only)	Slow blink	OFF	OFF (not supported)	
9.	Firmware upgrade in-progress	Fast blink	OFF	OFF (not supported)	

5 Other LEDs (ICP2-Portable)

##	*LED Name*	*Color*	*LED Behavior*
1.	POWER	**Blue**	ON
2.	LOW BATTERY	**Red**	• Battery is normal: OFF • Battery is low: ON • "GO" button is pressed & battery is very low: no programming is allowed, blinks in parallel to FAIL LED
3.	SERIALIZATION	**Green**	• Serialization = OFF: OFF • Serialization = ON & S/N (serial numbers) = OK: ON • Serialization = ON & no S/N: permanent slow blink • Serialization = ON, no S/N & "GO" button is pressed: no programming is allowed, blinks in parallel to FAIL LED
4.	COUNTER < 10	**Red**	• Counter < 10: ON • Counter = 0: permanent slow blink • Counter = 0 & "GO" button is pressed: no programming is allowed, blinks in parallel to FAIL LED
5.	Environment	**Green**	• Environment is selected & OK: ON

##	*LED Name*	*Color*	*LED Behavior*
	(1 of 6 LEDs)		• Environment is selected & invalid: permanent slow blink • Environment is invalid & "GO" button is pressed: no programming is allowed, blinks in parallel to FAIL LED
6.	PASS	**Green**	See above
7.	FAIL	**Red**	

6 Standalone Operation without PC

- Prepare an environment and transfer to programmer – see paragraph 14 for details
  NOTE: programmer is automatically ready for programming once the environment is saved in **non-volatile** memory
- Short pin GO to GND for at least 100ms to start programming
- Observe PASS/FAIL LEDs or/and pins PASS_OUT and FAIL_OUT – see paragraph 4 for details

7 Host Computer Requirements

- Pentium-4 or greater IBM PC compatible
- Windows-7/8/10/11. Contact Softlog Systems for operation with Win-95/98/NT/2000/XP/Vista/
- Display resolution: 1600x900 or higher
- 1GB of RAM
- At least 200MBytes of hard disk space
- Free USB port (all programmers), free RS-232 port (all programmers excluding ICP2-Portable) or free LAN connection (ICP2-COMBO only)

8 Installation

8.1 Important Note

- In the past the default directory was specified as "C:\Program Files\Soft-Log\...
- Starting from Jan-2015 ICP family software will be installed to a default directory C:\Softlog\ which allows to avoid virtual storage of CFG and INI files

8.2 Preliminary Installation

8.2.1 Software Installation

- Visit our site and get the latest software: http://www.softlog.com

8.2.2 Preliminary Hardware Installation

NOTE: USB driver installation is not required for operation with RS-232 port

- Install USB driver according to ***"USB Driver Installation.pdf"*** and ***"USB Driver (FTDI) Installation.pdf"*** instructions
  IMPORTANT:
  - disconnect ICP2 from USB
  - install the driver
  - connect ICP2 to USB
- Connect the programmer to its power supply (not required for ICP2-Portable and ICP2)
- Connect RS-232 or USB cable between PC and the programmer

8.3 ICP2-GANG Setup

- Install ICP2-GANG according to ***"ICP2-GANG Quick Start"*** document

8.4 ICP2-COMBO Setup

- Install ICP2-COMBO according to ***"ICP2-COMBO Quick Start"*** document
- Ventilation (fan) is **strongly** recommended if ICP2-COMBO(G3) is placed into a closed equipment

8.5 ICP2-Portable Setup

- Install ICP2-Portable according to ***"ICP2-Portable Quick Start"*** document

8.6 ICP2/ICP2(HC) Software Setup

- Run "ICP_Win.exe" Program
  - Double-click "ICP_Win" icon
  - Press "Yes" if message "Newer firmware is available. Upgrade now?" appears
- Run "Programmer/Quick Start Wizard" and follow the Wizard

8.7 FTDI Driver Latency

ICP2-Portable**(G3)** programmer operates with FTDI USB driver. Default latency of the driver is set to 16ms which slows down the programmer operation. ICP for Windows automatically tries to set the latency to 2ms, but it may not work. In this case change the latency as follows:

- Option 1 - run ICP for Windows as Administrator (click and hold right mouse button)
- Option 2 - change latency manually:
  - enter Control Panel → System → Device Manager → Ports (COM & LPT)
  - select USB Serial Port (COMx) → Port Settings → Advanced
  - change Latency Timer to 2ms (recommended). Note: 1ms is not recommended due to possible communication problems on several computers
  - press OK
  - restart your PC if it was prompted

9 Family-based Activation System

- Starting from July-2021 ICP2 G3 family programmers are activated by device/microcontroller family/algorithm
- Installed new software without activation will not work with non-Microchip/Atmel devices (for example STM, NXP, Nordic, etc.)
- Microchip/Atmel devices and serial EEPROM/flash memories will function without change
- You can always downgrade your existing ICP2 G3 programmers and use previous software/firmware
- Temporary activation is available for test purposes
- Contact us for more info: support@softlog.com

10 Checksum Calculation and Programming Buffers

ICP family software calculates HEX file checksum (CS) as follows (referred as "Checksum" or "ICP2 General Checksum"):

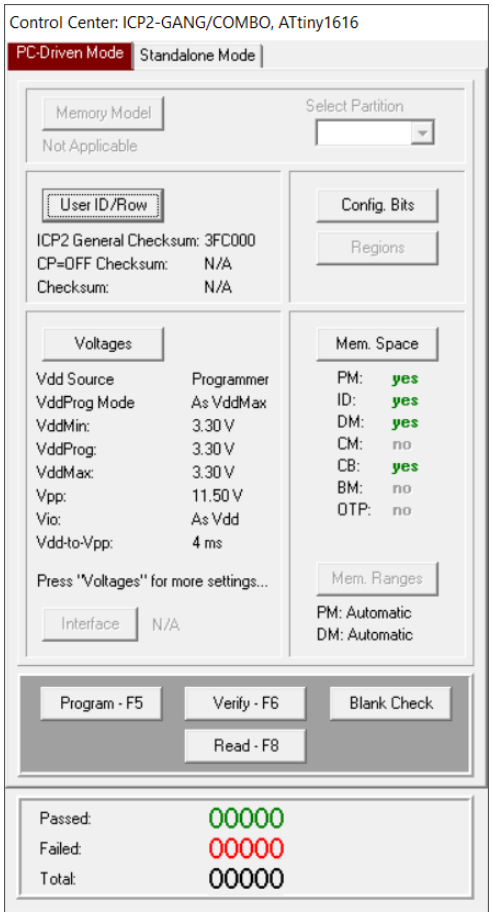
- Microchip®: Unprotected CS is calculated according to Microchip programming specifications (as MPLAB or MPLAB X)
- Other vendors or Microchip protected devices: in contrary to the programming specifications, **protected** CS is calculated as unprotected one. MPLAB/MPLAB X calculates it according to the specifications that makes the result CS nearly unusable (flash CS is not calculated at all)

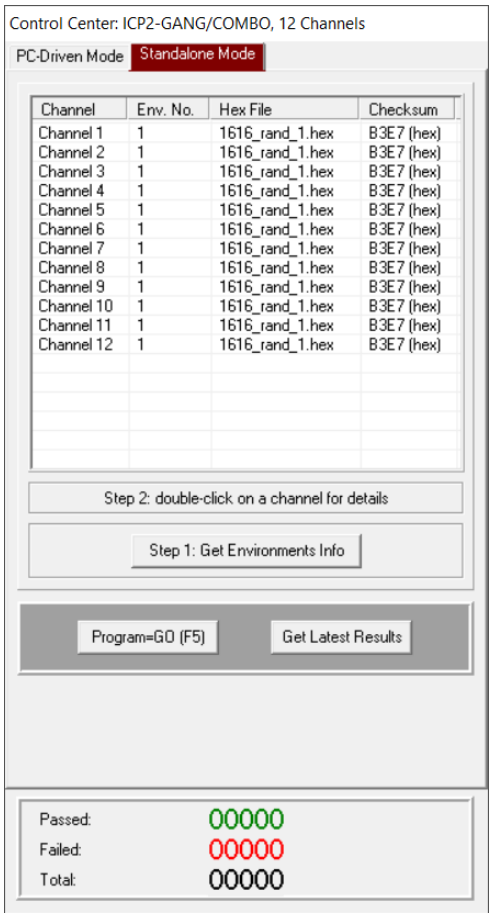
Starting from software 8.1.1 (Mar-2017) ICP for Windows displays "CP=OFF Checksum" and "Checksum" as they are expected to appear in MPLAB® IPE

Note that ICP software intentionally **doesn't** clear programming buffers automatically (it allows to merge partial HEX files), therefore CS may change if you load one HEX file after another. You can clear the buffers before loading by using Edit - Clear All Memory Buffers

11 Control Center

Control Center has 2 operation modes: PC-driven and Standalone





11.1 Control Center in PC-Driven Mode

Control Center in PC-driven mode allows the following operations:

- Select Memory Model and Partition if applicable for selected device
- Edit User ID/Row
- Edit Configuration Bits
- Set Regions or Security Bit if applicable for selected device
- Set voltages
- Select programming interface (for devices with dual interface, for example SWD or JTAG)
- Select memory spaces
- Set PM and DM ranges
- Programming, Verification, Blank Check and Read

11.2 Control Center in Standalone Mode

Control Center in standalone mode allows the following operations:

- Get environment information for all channels (button "Step 1: Get Environments Info")
- View environment details of selected channel (double-click on selected channel)
- Programming
- Get latest results

12 Menu Commands

12.1 File Menu (Alt-F)

Open (Import)...	Open a HEX file from disk and load it into buffer memory area
Save	Save the currently loaded file
Save As (Export)...	Save the buffer to a HEX file on disk
Universal HEX Loader	Load a HEX with virtual addresses (usually ATmega/ATtiny/AVR)
Load PM with zero offset	Load a HEX file with zero offset into PM (flash)
Import Data Memory	Load a HEX file with zero offset into DM (EEPROM)
Export Data Memory	Save DM into a HEX file with zero offset
Save Configuration	Save all current settings
Export Configuration (CFG) File…	Export and save current settings. **Note:** it also saves currently used icp01.cfg
Exit	Exit the software

12.2 Edit Menu (Alt-E)

Clear All Memory Buffers	Clear all memory buffers
Edit/Fill Program Memory	Fill area of Program Memory with a specified value
Edit/Fill Boot Memory	Fill area of Boot Memory with a specified value
Edit/Fill Data Memory	Fill area of Data Memory with a specified value
Edit/Fill OTP Memory	Fill area of OTP Memory with a specified value
Read-only Editors	Enable/disable edit the buffers above (disabled by default)

12.3 Environment (Alt-N)

Save Environment As…	Save current setup and buffers to an environment file (*.pj2)
Transfer Environment to Programmer…	Transfer environment (*.pj2) to programmer
Transfer Different Environments to ICP2-GANG/ICP2-COMBO	Transfer different PJ2 files to programmer in single dialog
Environment Wizard…	

12.4 Serialization Menu (Alt-S)

Disable	Disable serialization
Load File	Load serialization file
Create File	Create serialization file

12.5 Device Menu (Alt-D)

Select by Name…	Select a type of device to be programmed

12.6 Programmer Menu (Alt-P)

Select Programmer	Select programmer
GANG/COMBO Configuration	Select active GANG/COMBO channels (64 max)
Assign Address and Baud Rate to GANG/COMBO Box	Assign box address to currently connected programmer (1-16), see ***"ICP2-GANG Quick Start"*** for details
Quick Start Wizard	

12.7 Run Menu (Alt-R)

Program	PC-driven mode: program data from the buffer(s) into the device Standalone mode: activate standalone programming
Verify	Verify data in the device against the data in the buffer(s)
Blank Check	Check data in the device for the blank state
Read	Read device and store the data in the buffers

Program Only	Open a window for repeated programming

12.8 Communication Menu (Alt-C)

RS-232/USB/LAN COM	Select the desired COM port and connect
Connect	Connect to the programmer

12.9 Options Menu (Alt-O)

Voltage	Set desired voltages
Clock/Data/MCLR(Advanced)	Set desired advanced voltage parameters
Preferences	Select programming preferences
Firmware Upgrade	Execute firmware upgrade
Summary of Options/Families, Collect My Firmware IDs	View summary of options/families, save Firmware IDs for activation
Activate Options/Families…	Execute activation of options/families - Families - DLL/Command Line (D) - Secure Programming Support (S) - Microchip only (K): Keeloq® - Microchip only (P): 16-bit (dsPIC®/PIC24) - Microchip only (X): 32-bit (PIC32, ATSAM) Note: **Contact Softlog Systems for activation details**

12.10 Speed Optimization Menu (Alt-T)

Run Speed Optimization Utility	Run a utility which automatically configures the programmer settings for the optimal speed performance
Optimization Summary	Show optimization summary

12.11 Help Menu (Alt-H)

Read me	Display "Readme_w.txt" file
About	Connect to the programmer and display software and firmware versions

13 Shortcuts

Save	Ctrl-S
Open (Import)...	Ctrl-O
Program	F5
Verify	F6
Blank Check	F7
Read	F8
Programming Only	F9

14 Preparing and Transferring Environment to Programmer

14.1 Prepare and Transfer Environment with Wizard

- Run "Environment/Environment Wizard" and follow the Wizard
- Select programmer and press "Next"
- ICP2-GANG/ICP2-COMBO only: select GANG channels and press "Next"
- Select environment number and press "Next"
- Select Device
  Select a device to be programmed and press "Next"

- Set Voltages (and Advanced button) and press "Next"

- Set Preferences and press "Next"

- Load (open) a HEX file.
  NOTE: The programmer software is able to read User ID, data memory (EEPROM), OTP and configuration bits from the HEX file

- Load serialization (SER) file (if required) and press "Next"

- Save Environment
  - Press on "…" button
  - Type in environment name, 16 characters max
  - Press "Save"
  - Press "Next"

- Transfer Environment to Programmer
  - Press on "Transfer Environment…" button, select your environment file and press "Open"
  - Wait until environment is transferred to all channels
  - Press "Next"

- Switch to Standalone Mode
  - Press on "Standalone Mode" button
  - Press "Finish"

- Your system is ready for standalone programming

- View the transferred environment as shown in Paragraph 15

14.2 Prepare and Transfer Environment Manually

Some settings (memory areas, memory, etc.) are not available with the Environment Wizard therefore manual procedure below is recommended in several cases

- Validate that your part number (device name) is selected correctly

- Change voltages, clock frequency and other parameters if required

- Load your HEX file

- Change memory space (example): Control Center → select PM and CB only

- Load serialization file (if required): Serialization → Load File

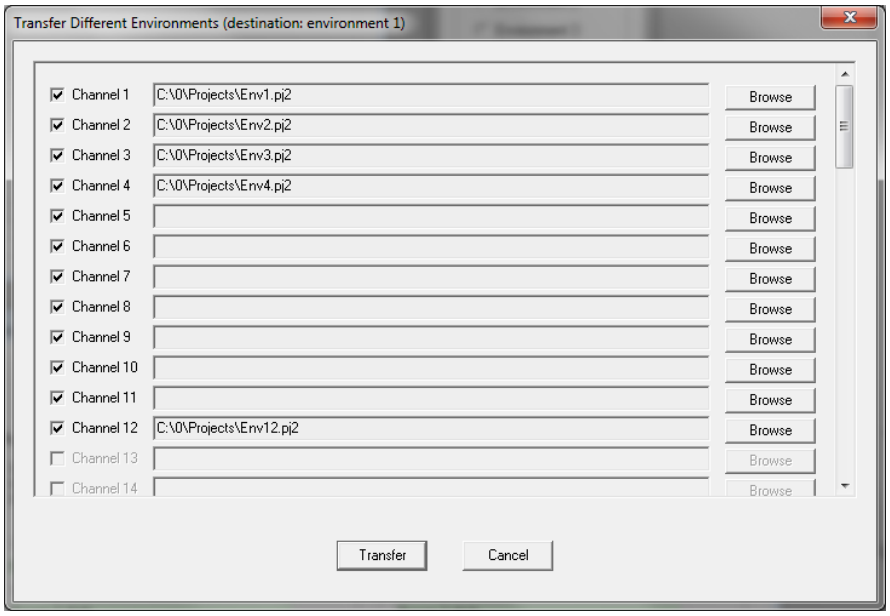
- Save environment: Environment → Save Environment As…

- Transfer the saved environment file PJ2: Environment → Transfer Environment to Programmer…

14.3 ICP2-GANG and ICP2-COMBO: Transfer Different Environments to Different Channels

14.3.1 Option 1: Multiple channel mode, single operation

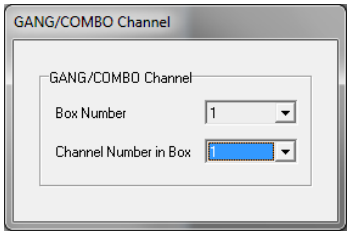
- Create different environment files *.PJ2, for example Env1.pj2, Env2.pj2, Env3.pj2, Env4.pj2, …, Env12.pj2 as specified in 14.2 (don't transfer)
- Select Environment → Transfer Different Environments to ICP2-GANG/ICP2-COMBO and load PJ2 files (Env5.pj2…Env11.pj2 are not shown)



- Press "Transfer"

14.3.2 Option 2: Single channel mode, multiple operations

- Create different environment files *.PJ2, for example Env1.pj2, Env2.pj2, Env3.pj2, Env4.pj2, …, Env12.pj2 as specified in 14.2 (don't transfer)
- Select single channel mode: Programmer → Select Programmer → ICP2-GANG(Single) / ICP2-COMBO(Single)
- On the pop-up window select desired channel, for example channel 1

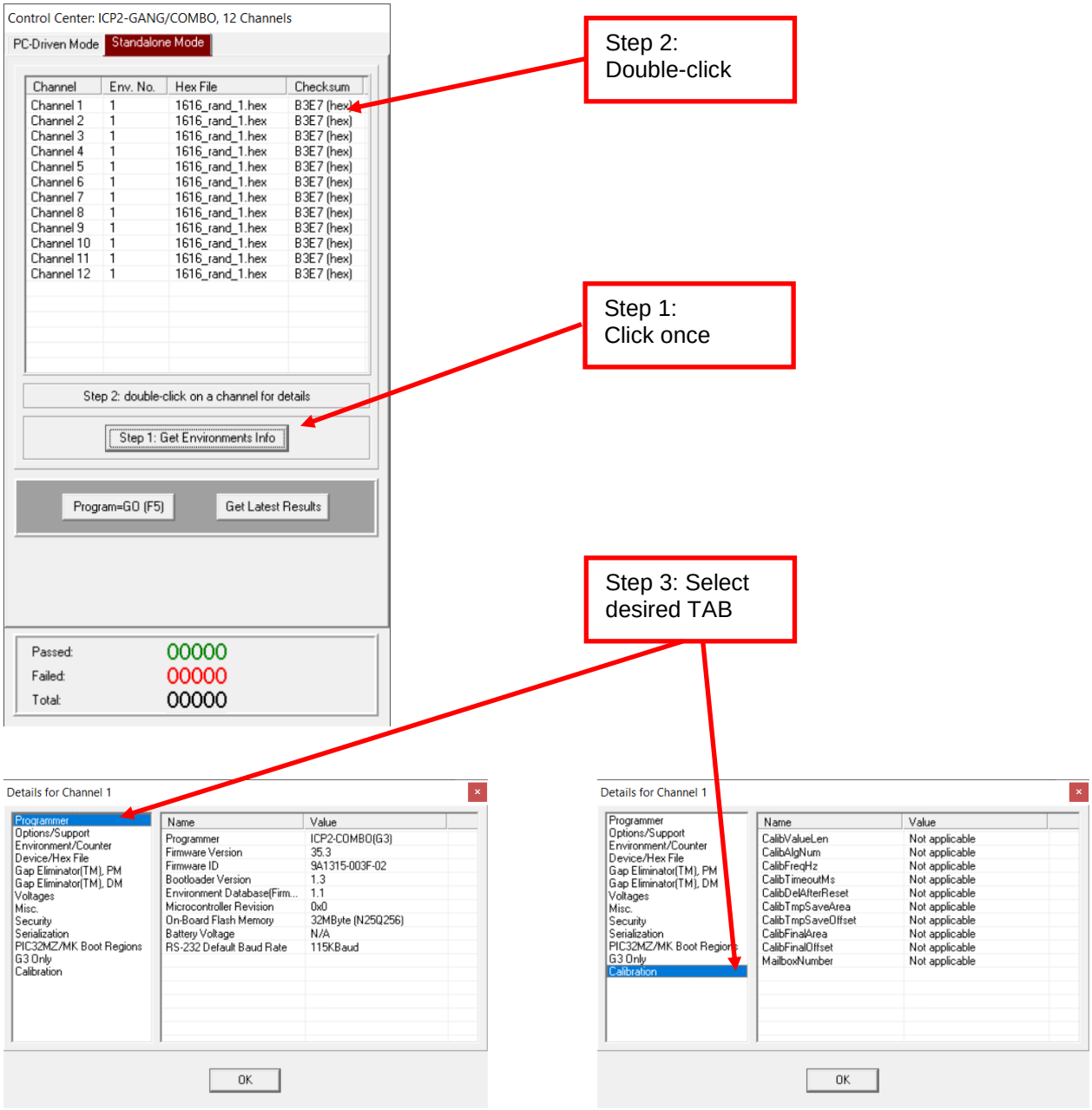


- Transfer Env1.pj2: Environment → Transfer Environment to Programmer…
- Repeat 2 steps above for channels 2, 3, 4…12 with Env2.pj2, Env3.pj2 and Env4.pj2… Env12.pj2
- Restore GANG/COMBO selection: Programmer → Select Programmer → ICP2-GANG/ICP2-COMBO

15 Viewing Environment

- Preparation: select desired environment number

- Active environments inside the programmer can be viewed as shown below

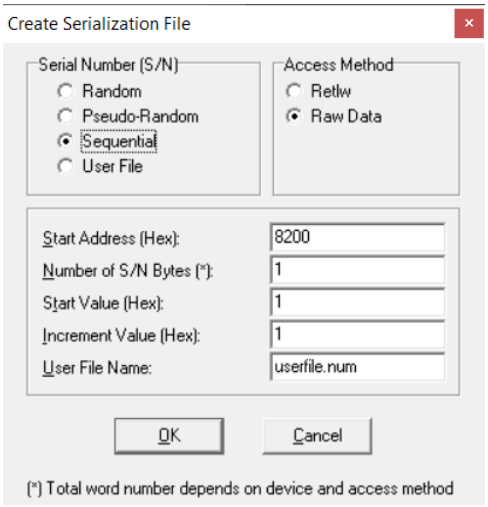


16 Serialization

16.1 Create Serialization File

- Select "Serialization/Create File" to generate a serialization file

- Enter the following data:
  - Serial Number (serialization scheme): random, pseudo-random, sequential and user file
  - Start Address. The address should be valid for the device
  - Number of S/N Bytes. Enter number of bytes (1 to 8) for your serial number
    WARNING: Total number of result **words** (addresses) depends on the device and access method
  - Start Value. Enter the start value (1 to 16 hex digits). If the start value is greater than the maximum value for the number of bytes selected the most significant digits will be truncated. The start value must differ from zero for pseudo-random scheme.
  - Increment Value. Valid for the sequential scheme only
  - User File Name. Valid for "user file" scheme only
  - Access Method. Select Retlw or Raw Data



- Press OK to save a serialization file

NOTES:
- A currently selected serialization file will be updated after any successful programming for single-channel programming and after any programming attempt for ICP2-GANG/ICP2-COMBO
- The "retlw" opcode ("retlw" access method) will be automatically generated for a selected type of devices, i.e.:
  - 08(Hex) for low-end microcontrollers (12C5xx, etc)
  - 34(Hex) for mid-range microcontrollers (16C/Fxxx)
  - b6(Hex) for high-end microcontrollers (17C7xx)
  - 0c(Hex) for enhanced microcontrollers (18Fxxx)
  - 054(Hex) for 16-bit devices (pattern: 0000_0101_0100_kkkk_kkkk_dddd)

16.2 Serialization File Example 1

SerializationScheme = 2 (0-Random, 1-Pseudo-Random, 2-Sequential, 3-User File)
StartAddress = 0005 (Hex)
NumberOfWords = 2 (Hex)
CurrentValue = 0000000000001234 (Hex)
IncrementValue = 1 (Hex)
UserFile = userfile.num
AccessMethod = 0 (0-retlw,1-raw data)

The following program memory locations will be updated as follows:

- PIC16xxx
  0005: 3434
  0006: 3412

- PIC12C5xx
  0005: 0834
  0006: 0812

16.3 Serialization File Example 2 (User File Scheme)

SerializationScheme = 3 (0-Random, 1-Pseudo-Random, 2-Sequential, 3-User File)
StartAddress = 0005 (Hex)
NumberOfWords = 2 (Hex)
CurrentValue = 0000000000001234 (Hex)
IncrementValue = 1 (Hex)
UserFile = File1.num
AccessMethod = 0 (0-retlw, 1-raw data)

User file should contain serial numbers in HEX radix, for example:
1111
2222
3333
4444
5FC1

The user file will be updated by placing semicolon (;) at very beginning of the string, for example:
; 1111
; 2222
3333
4444
5FC1

If your numbers start from very beginning of the string the 1-st digit will be replaced by semicolon:
;111
;222
3333
4444
5FC1

16.4 Enable Serialization

Select "Serialization/Load File" to activate serialization

16.5 Disable Serialization

The serialization will be disabled in the following cases:
- "File/Open..." command is executed
- "Serialization/Disable" command is executed
- "Edit/Read-only Editors" is set to edit mode
- "Run/Read" command is executed
- a new device is selected
- user's serialization file is empty
- Control Center switches between PC-driven and standalone modes

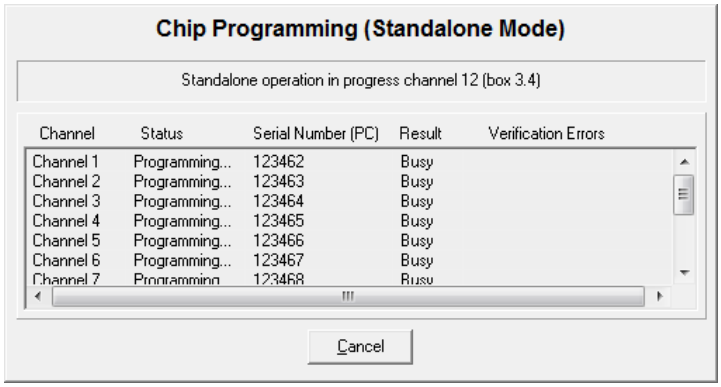
16.6 Standalone Serialization

- Make all settings (select device, voltages, etc.)
- Load a HEX file
- Select "Serialization/Load File" to activate serialization
- Create an environment by "Environment/Save Environment As…"
- Transfer the environment to programmer

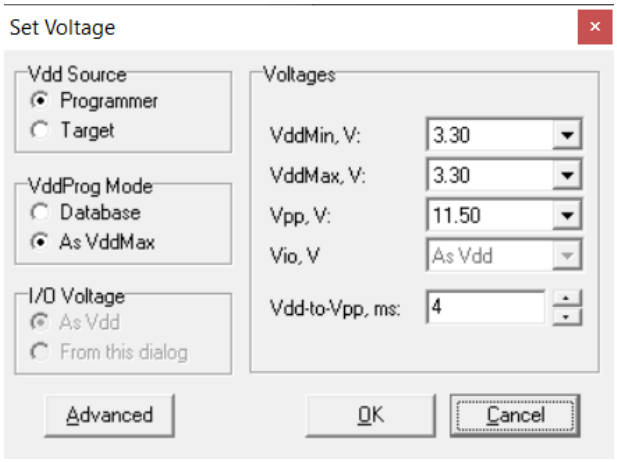
16.7 PC-Driven Serialization in Standalone Mode

PC-driven serialization in standalone mode may be required for GANG/COMBO

- Switch Control Center to Standalone mode
- Validate that all required channels are enabled – see 15
- Select "Serialization/Load File" to activate serialization
- Execute programming (F5)



17 Options → Voltages



17.1 Vdd Source and VddProg Mode

The programmer executes operations at the following Vdd voltages

##	*Vdd Source*	*VddProg Mode*	*Vdd during Programming*	*Vdd during Verify*	*Vdd during Blank Check*	*Vdd during Read*
1.	Programmer	Database	Database	VddMin, VddMax ***(Note 2)***	VddMin	Database
2.	Programmer	As VddMax	VddMax ***(Note 1)***	VddMax	VddMax	VddMax
3.	Target	Any	Target	Target	Target	Target

Notes:
1) Use "As VddMax" mode if you want to change default programming voltage
2) Set VddMin=VddMax to disable the 2-nd verification pass

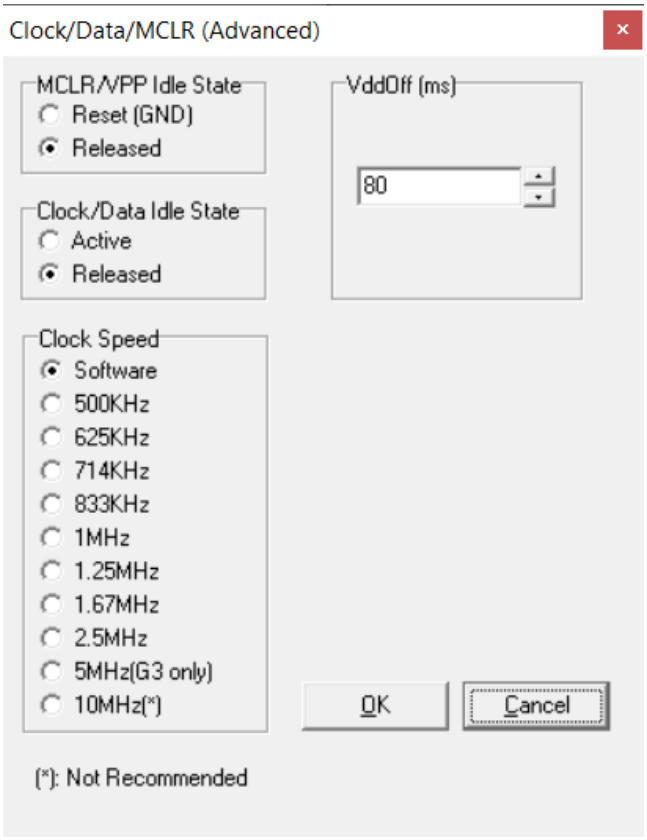
17.2 Vpp Voltage

The Vpp voltage is the same for all the operations

17.3 Vdd-to-Vpp Delay

Delay between Vdd and Vpp can be in range 0.1...250ms. It is recommended to use default delay of 4ms to correctly enter the programming mode. Longer delays may be useful if the Vdd line has high capacitance (more than 200uF) which causes the Vdd to rise slowly

18 Options → Clock/Data/MCLR(Advanced)



18.1 MCLR/VPP Idle State

Reset (GND): Programmer permanently keeps MCLR in reset state (GND) when no operation
Released: Programmer releases MCLR with weak pull-down of about 160KΩ

18.2 Clock/Data Idle State

Active: Programmer configures data/clock pins as outputs when no operation
Released: Programmer releases data/clock with weak pull-downs of about 300KΩ

18.3 Clock Speed

10MHz is not recommended

18.4 VddOff Delay

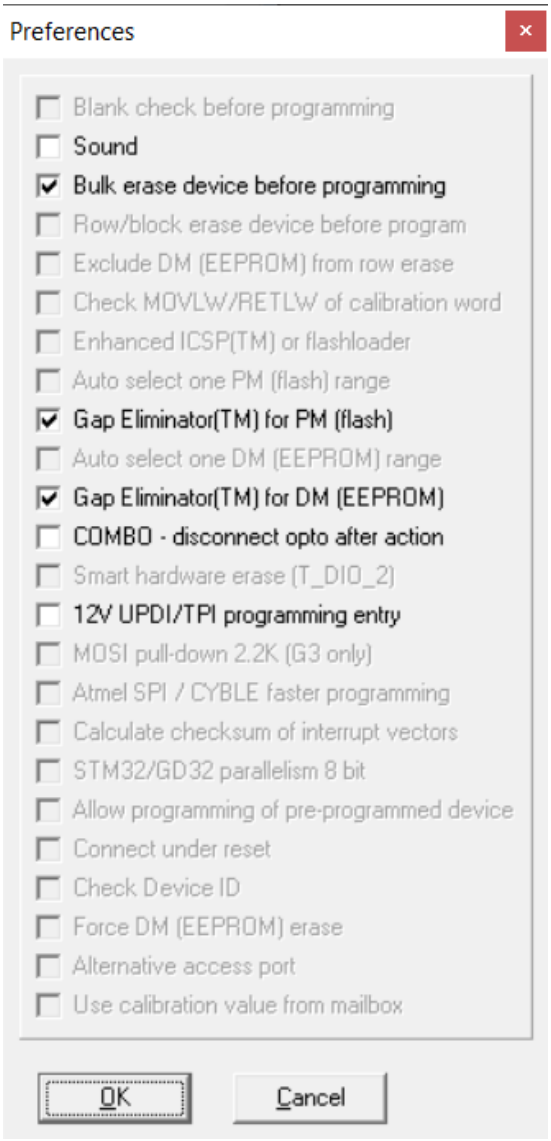
VddOff delay is required for Vdd discharge, range 5-250ms (default 80ms)

18.5 RESET Pulse after Operation

"RESET Pulse after Operation" is not a single menu item, it's provided as a result of combined settings as follows (**all** settings **together**):
- Vdd source = **Target**
- MCLR/VPP Idle State = **Released**
- Clock/Data Idle State = **Released**

19 Options → Preferences

NOTE: some items are grayed out if they are not supported by a selected device



19.1 Blank check before programming [ ]

Enables/disables blank check before device programming. This option is not useful for flash devices

19.2 Bulk erase device before programming [x]

When the option is ON the device will be automatically erased by bulk/chip/mass erase mechanism
HIGHLY IMPORTANT:
- this option must be set to ON for proper operation with most of devices
- usually it's the only option to erase code protected device

19.3 Row/block erase device before programming [ ]

When the option is ON the device will be automatically erased by the row/block/sector erase mechanism
IMPORTANT:
- row erase usually can't erase a code protected device
- available for several devices

19.4 Exclude DM (EEPROM) from row erase [ ]

When the option is ON the DM (EEPROM) is excluded from row erase procedure

NOTES:
- available for several devices, may be useful to preserve EEPROM
- DM (EEPROM) space should be **excluded** from operation (see Memory Space on Control Center)

19.5 Check MOVLW/RETLW of calibration word [x]

When the option is ON an opcode of the calibration memory is tested during programming
NOTE: available for devices which have a calibration word with movlw/retlw opcode (PIC12F519, PIC12F675, etc.)

19.6 Enhanced ICSP™ Programming or Flashloader [x]

When the option is ON the device is programmed/verified faster using Enhanced ICSP™ method or via a flashloader

IMPORTANT: external or internal pull-down resistor is required for Enhanced ICSP™ of dsPIC33/PIC24 devices

IMPORTANT - Enhanced ICSP™ limitations (Microchip® silicon issues):
- PGEC3/PGED3 programming pair does not work on several devices – check Microchip® errata
- Enhanced ICSP™ may not work if "Windowed WDT" is enabled

19.7 Auto select one PM (flash) range [ ]

Obsolete preference, overridden by "Gap Eliminator for PM"

19.8 Gap Eliminator™ for PM (flash) [x]

When this option is ON ICP family programmer automatically excludes **multiple** empty (blank) PM areas in the HEX file from the programming process, resulting in shortened programming time. See paragraph 20 for more details

19.9 Auto select one DM (EEPROM) range [ ]

Obsolete preference, overridden by "Gap Eliminator for DM"

19.10 Gap Eliminator™ for DM (EEPROM) [x]

When this option is ON ICP family programmer automatically excludes **multiple** empty (blank) DM areas in the HEX file from the programming process, resulting in shortened programming time. See paragraph 20 for more details

IMPORTANT: Due to different silicon read/write protection mechanisms the DM (EEPROM) may be not erased before programming for several devices, therefore test your device before going to production. If it is then don't enable this feature

19.11 COMBO: disconnect opto after action [ ]

When the option is ON the ICP2-COMBO opto-relay barrier will be disconnected after programming/verification/blank check/read

19.12 Smart hardware erase (T_DIO_2) [ ]

Applicable to devices with hardware erase pin (currently Cortex M7 SAM E/S/V). If the option is ON then hardware erase is executed if no communication with the device is established (if communication is OK then bulk or row/block erase is done)

19.13 12V UPDI/TPI programming entry [ ]

When the option is ON 12V entry method is used for devices with UPDI or TPI interface.
IMPORTANT: validate that your hardware is designed for 12V on UPDI or MCLR pin (actual voltage level is defined by Vpp setting)

19.14 MOSI pull-down 2.2K (G3 only) [ ]

When the option is ON the internal pull-down resistor 2.2K on T_MOSI is enabled, useful for Enhanced ICSP™

19.15 Atmel SPI / CYBLE faster programming [x]

When the option is ON Atmel SPI devices (ATmega, ATtiny, etc.) are programmed faster by pre-programming fuse bits with maximum oscillator speed configuration. Final fuse bits are reprogrammed at the end of the operation. Higher baud rate is used for Cypress CYBLE modules

19.16 Calculate checksum of interrupt vectors [ ]

Applicable to LPC80x devices only

19.17 STM32 parallelism 8 bit [ ]

When the option is ON STM32F2xx/4xx/7xx devices are erased and programmed by 8-bit (x8) parallelism instead 32-bit one (x32). x8 algorithm is much slower but it's usually required for VDD < 2.7V. Note: 16-bit (x16) parallelism is not supported

19.18 Allow programming of pre-programmed device [ ]

Allows programming of pre-programmed devices (applicable to Nordic nRF51)

19.19 Connect under reset [ ]

Applicable to several STM32 devices

19.20 Check Device ID [x]

Checks Device ID before programming

19.21 Force DM (EEPROM) erase [ ]

Forces DM erase even for devices with fuse "Preserve EEPROM". Applicable to Atmel SPI devices

19.22 Alternative access port [ ]

Applicable to several STM32 devices

19.23 Use calibration value from mailbox [ ]

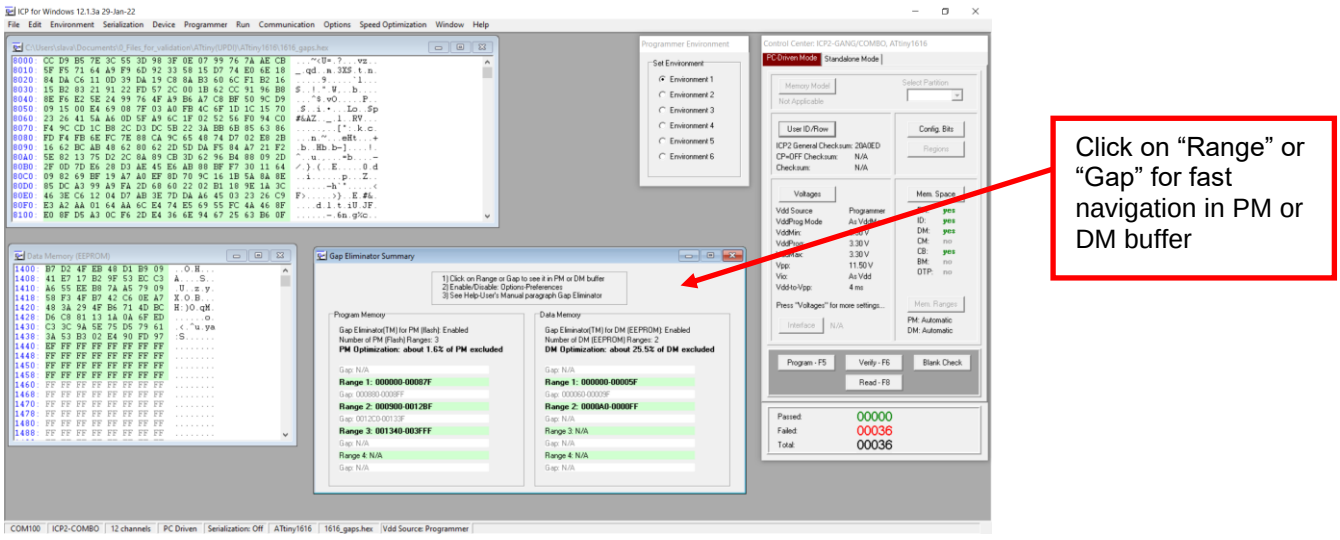
Applicable to ATmega devices when frequency calibration is required

20 Gap Eliminator™

20.1 Overview

The Gap Eliminator™ enables end customers to exclude empty (blank) flash and EEPROM areas in the HEX file from the programming process, resulting in shortened production cycles and reduced manufacturing costs. This powerful feature is available in all of Softlog's in-circuit programmers

Starting from Jan-20 permanent "Gap Eliminator Summary" window is available:



20.2 How It Works

In addition to the critical data they carry, HEX files may also contain multiple empty areas (gaps). These gaps may come at the beginning, in the middle, or at the end of the HEX file. Thus, when programming a microcontroller, the empty bytes of a HEX file are also burned onto the microcontroller. In order to "skip" these gaps, a typical programmer usually allows the operator to define a single programming range, thus enabling two empty areas to be skipped at the beginning and end of the file. However, if the gap(s) are located between valid data areas (see example below), this is not an effective solution.

Softlog's Gap Eliminator™ feature solves this problem. Before a production run, it automatically analyzes the HEX file and effectively removes multiple gaps (up to five) from the Program Memory (flash) and Data Memory (EEPROM). This significantly reduces programming time for mass production operations.

20.3 Example of HEX File with Gaps

Memory size = 4096 bytes (address range 0x0000…0x0FFF)

Gaps are highlighted in yellow; **valid data** in green

```
Address 0x0000…0x0007: FF FF FF FF FF FF FF FF
…
Address 0x0078…0x007F: FF FF FF FF FF FF FF FF
Address 0x0080…0x0087: 01 53 A4 67 88 A5 CD 6F
Address 0x0088…0x008F: 01 23 45 67 89 AB CD EF
…
Address 0x01F8…0x01FF: 51 F3 45 F7 89 A6 CC CF
Address 0x0200…0x0207: FF FF FF FF FF FF FF FF
Address 0x0208…0x020F: FF FF FF FF FF FF FF FF
…
Address 0x07F8…0x07FF: FF FF FF FF FF FF FF FF
Address 0x0800…0x0807: 01 53 A4 67 88 A5 CD 6F
Address 0x0808…0x080F: 01 23 45 67 89 AB CD EF
…
Address 0x09F8…0x09FF: 51 F3 45 F7 89 A6 CC CF
Address 0x0A00…0x0A07: FF FF FF FF FF FF FF FF
Address 0x0A08…0x0A0F: FF FF FF FF FF FF FF FF
…
Address 0x0FF0…0x0FF7: FF FF FF FF FF FF FF FF
Address 0x0FF8…0x0FFF: 01 53 A4 67 88 A5 CD 6F
```

As noted, a typical programmer allows you to define one range 0x0080…0x0FFF (3968 bytes), skipping the empty area at the beginning of the file. **This reduces the size of the HEX file by 128 bytes (3.2%).**

Using a Softlog ICP programmer with the Gap Eliminator™ feature, three programming ranges can be defined for this example:

- 0x0080…0x01FF (384 bytes)
- 0x0800…0x09FF (512 bytes)
- 0x0FF8…0x0FFF (8 bytes)

This effectively eliminates all the gaps and **reduces the size of the HEX file to be programmed by 3,192 bytes (77.9%)**.

21 Speed Optimization Utility

The Speed Optimization Utility is a Wizard that guides the user through the ICP programmer configuration settings to ensure optimal speed performance. These settings include Clock/Data speed, Vdd-to-Vpp delay, VddOff delay, Gap Eliminator, Enhanced ICSP™ and more.

Enter "Speed Optimization → Run Speed Optimization Utility"

22 Configuration File

The ICP setup is saved in a configuration file named "icp01.cfg".

IMPORTANT: the program reads a configuration file that is located in a directory which specified in "Start in" property. This approach allows creation of unlimited configurations on the same PC

Normally, a configuration file should be **not** modified by a text editor.

23 ICP for Windows: Limited Command Line Parameters

Some parameters can be loaded from the command line:

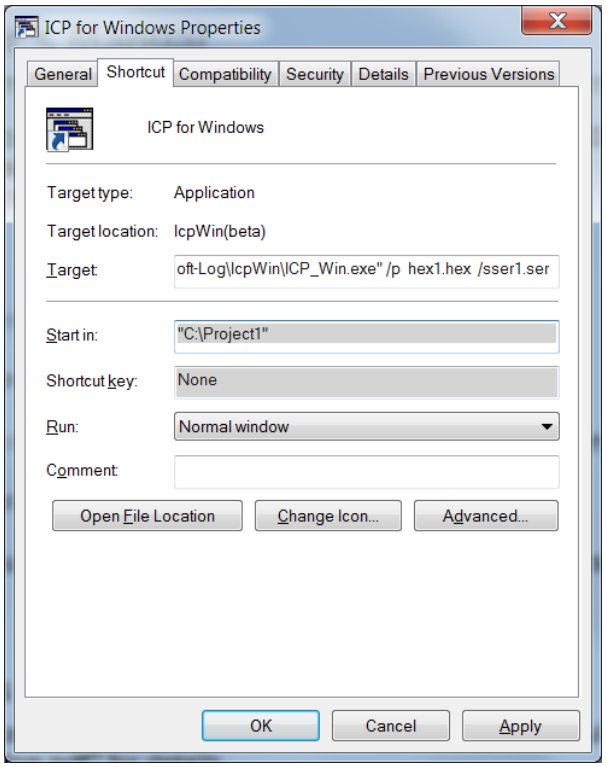
<Hex file> - hex file to be loaded
/c<Configuration file> - configuration file to be loaded, overwrites local "icp01.cfg"
/s<Serialization file> - serialization file to be loaded
/p - production mode (one-touch operation)

Examples:

- Start in the production mode and load file "hex1.hex":
  <path to ICP_Win.exe> /p hex1.hex

- Start in the production mode and load hex file "hex1.hex" and serialization file "ser1.ser":
  <path to ICP_Win.exe> /p hex1.hex /sser1.ser

  "Start in" property should specify a directory where "hex1.hex" and "ser1.ser" are located



24 DLL Functions

ICP family programmers can be run from the user's application using powerful set of DLL functions. See document ***"DLL Description.pdf"*** for details

25 Command Line Interface (non-GUI)

ICP family programmers can be run from the user's application using full-featured command line interface. See document ***"ICP Command Line.pdf"*** for details

26 Secure Programming

Your hex files contain business-critical intellectual property that could be compromised during the contract manufacturing process. Our Secure Programming feature provides several layers of protection that dramatically reduce the risk of unauthorized reconstruction of hex data. See document ***"Secure Programming Utility User's Manual.pdf"*** for more details

27 Standard LAN Configuration

NOTE: see ***"ICP2-COMBO LAN Configurator User's Manual.pdf"*** for detailed description of all features

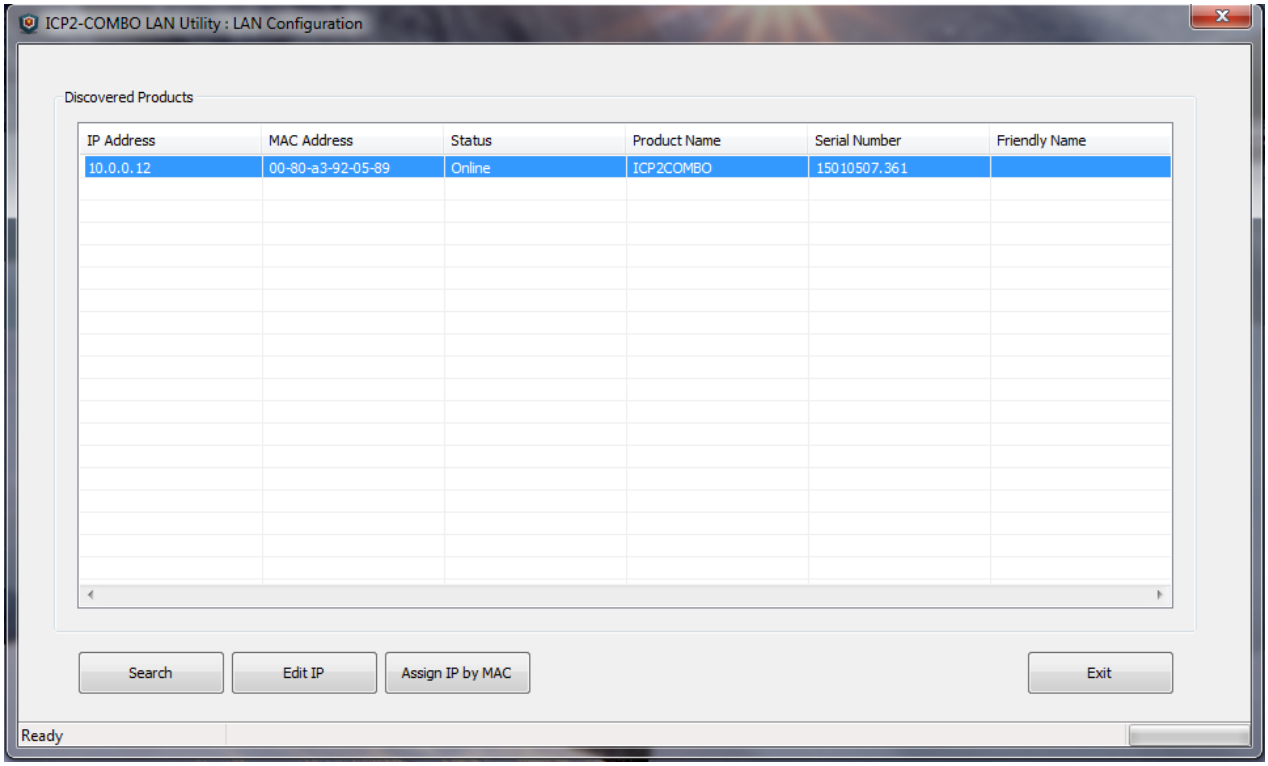
To install and run the software supplied, follow the steps below:

- Run "IcpLan_setup_x_x_x_MMM-YYYY.exe" and follow the on-screen instructions
- Connect ICP2-COMBO programmer to the same Ethernet network as your PC:
  - use straight cable for connection via network hub or switch
  - use crossover cable for direct connection to the PC
- Run "ICP-LAN" application that is located under C:\Softlog\IcpLan. An opening screen appears:

The "ICP-LAN" application provides 3 configuration features:
- "LAN Configuration" which discovers ICP2-COMBO programmers and changes IP network parameters
- "COM Port Redirection" which creates a virtual COM (CPR) port for ICP2-COMBO programmer
- "Create Remote Port" which creates remote port w/o discover procedure

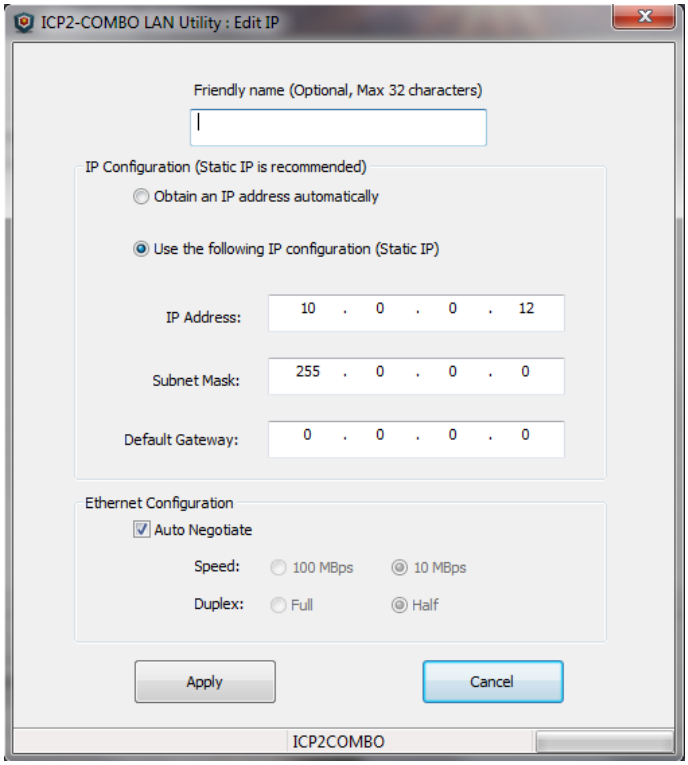
27.1 LAN Configuration

- Click "LAN Configuration" button
- The application automatically starts discovering programmers. You can retry by clicking "Search" button
- The following screen appears:



NOTES:
- verify that detected product serial number is the same as printed on the ICP2-COMBO sticker
- if your device was not found refer to "Advanced IP Configuration" below (paragraph **Error! Reference source not found.**)
- if status of discovered device is "Unreachable", reconnect the power to the device and retry searching, otherwise refer to "Advanced IP Configuration".

- Select (highlight) the discovered ICP2-COMBO programmer
- Click "Edit IP" button. The following screen appears:



- Change your settings if required

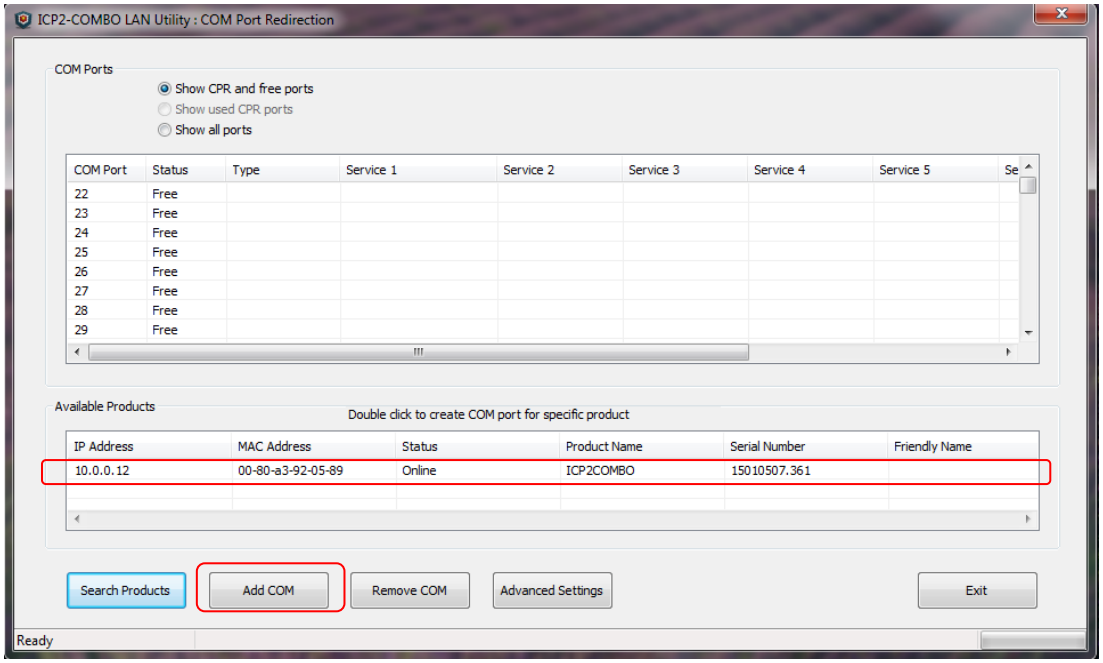
  NOTES - the following settings are recommended:
  - assign a friendly name to ICP2-COMBO, for example "COMBO-12 for tester 4"
  - use static IP for convenient operation with a virtual CPR COM port

- Press "Apply". Wait until operation is complete (may take about 1 minute)

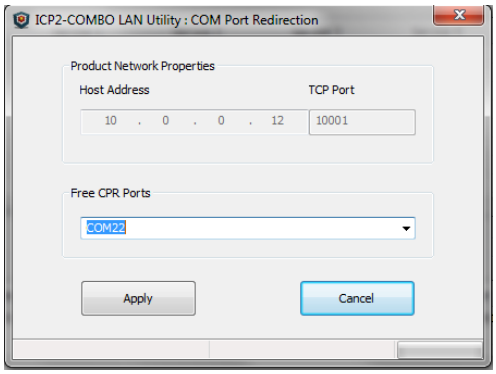
- Press "Exit"

27.2 COM Port Redirection

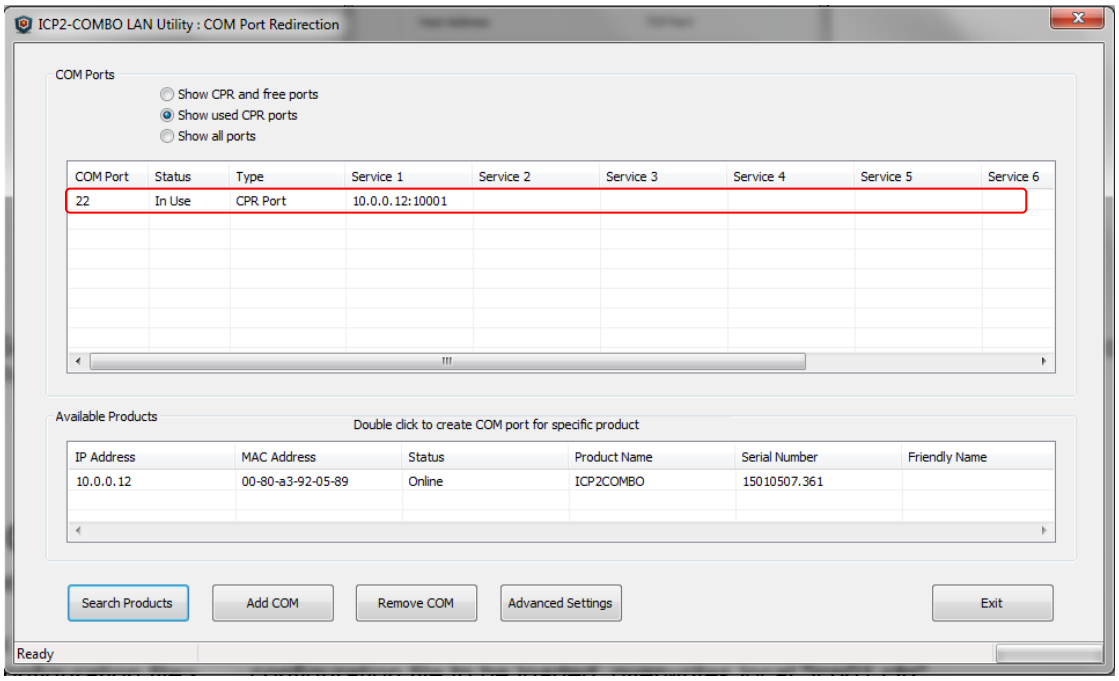
- Click "COM Port Redirection" button (from the opening screen). The following screen appears:



- Select (highlight) the discovered ICP2-COMBO programmer on the bottom list
- Click "Add COM" button. The following screen appears:



- Select desired COM port number and press "Apply". COM Ports list on "COM Port Redirection" window will refresh automatically:



- Press "Exit"

28 Manual Production Mode (One-Touch Operation)

The production mode is a powerful option for volume programming

The following steps should be done to correctly prepare the software for programming in the production mode:

- Create a subdirectory (C:\FILE_HEX)
- Copy your CFG, HEX and SERIALIZATION files to FILE_HEX subdirectory (for example: "hex1.hex" and "ser1.ser")
  NOTE: serialization file is optional
- Change ICP_Win shortcut property "Start in" to C:\FILE_HEX
- Change ICP_Win shortcut property "Target" to C:\...\ICP_Win.exe hex1.hex /sser1.ser /p
- Double-click ICP_Win icon for programming

The program will be terminated in the following cases:
- Communication error
- Hex file error
- Serialization file should be loaded (/s appears) but loading is failed

29 In-Circuit Programming (Electrical Parameter Summary)

NOTE: for more details see ***"ICP2-GANG(G3) Specification.pdf"***, ***"ICP2-COMBO(G3) Specification.pdf"***, ***"ICP2(G3) Specification.pdf" and "ICP2-Portable(G3) Specification.pdf"***

29.1 VDD

- Maximum Vdd current consumption by the application circuit:
  - ICP2/ICP2-GANG/ICP2-COMBO: 250mA per channel
  - ICP2-Portable: 100mA

- Maximum Vdd capacitance: 1000-10000uF. For ICP2/ICP2-GANG/ICP2-COMBO, increase Vdd-to-Vpp delay by about 20ms for every 1000uF

NOTE: If your circuit has low current consumption (less than 10mA) in conjunction with high capacitance (more than 100uF), a load resistor of 100-510 Ohm must be connected between Vdd and GND pins of the programmer for faster discharge of Vdd capacitor

29.2 VPP

VPP recommended load: > 1KOhm, < 22nF

WARNING: due to high VPP requirements ICP2-Portable may be not suitable for old OTP devices

29.3 T_MOSI, T_MISO, T_DIO_0/1/2

Recommended load: > 3.3KOhm, < 33pF

29.4 Delay between VDD and VPP

Default 4ms is recommended, it should be increased for big Vdd capacitors to allow full Vdd charge before Vpp rise – see 29.1

30 Target Cable

Softlog Systems recommends using the following cable between ICP family programmer and target device:

- Length: as short as possible, less than 50cm
- Structure: unshielded separate wires
- Low-pass filter: 22-47pF between CLOCK and GND **as close as possible** to the target microcontroller is recommended, especially for long cables (> 20cm)

31 Appendix A: Power Supply

ICP family programmers (excluding ICP2-Portable) are shipped with their own power supplies. If the user wishes to connect his/her own power, make sure the following specifications are met:

Programmer	*Output Voltage*	*Output Current*	*Center Terminal, 2.1mm*	*Recommended Power Supply*
ICP2	12VDC	0.5A	"-" or "+"	GST25A12-P1J (Mean-Well)
ICP2(HC)	12VDC	1.5A	"-" or "+"	
ICP2-Portable	12-15VDC	0.5A	"-" or "+"	
ICP2-GANG	12VDC	1.5A	"+"	
ICP2-COMBO	12VDC	5A	"+"	GST60A12-P1J (Mean Well)

32 Appendix B: Return Values (Errorcodes)

```
enum AUTO_ERROR_LEVEL { //return values
       AUTO_OK                    = 0,  //operation OK
       AUTO_DB_ERR                = 1,  //database error
       AUTO_COM_ERR               = 2,  //communication error
       AUTO_VDD_ERR               = 3,  //Vdd overload error
       AUTO_VPP_ERR               = 4,  //Vpp overload/injection error
       AUTO_HEX_ERR               = 5,  //HEX file loading error
       AUTO_SER_ERR               = 6,  //serialization file error
       AUTO_VER_ERR               = 7,  //verification error
       AUTO_ERR_NO_SPACE          = 8,  //no space/area selected for operation
       AUTO_SAVE_ERR              = 9,  //file save error
       AUTO_SOCK_ERR              = 10, //socket communication error (obsolete)
       AUTO_I2C_ERR               = 11, //Connection error with UUT or GANG/COMBO address conflict
       AUTO_DLL_ERR               = 12, //DLL programming is not activated
       AUTO_KEY_ERR               = 13, //key generation error (KeeLoq)
       AUTO_CFG_ERR               = 14, //configuration (CFG) file error
       AUTO_COM_NUM_ERR           = 15, //invalid COM number
       AUTO_COM_BUSY_ERR          = 16, //selected COM is busy
       AUTO_COM_BAUD_ERR          = 17, //invalid baud rate
       AUTO_COM_NO_OPEN           = 18, //can't open COM port
       AUTO_USER_CANCEL           = 19, //user cancel
       AUTO_IN_PROGRESS           = 20, //operation in progress
       AUTO_BC_ERR                = 21, //blank check error
       AUTO_OP_NOT_ALLOW          = 22, //operation not allowed for selected programmer
       AUTO_FW_INVALID            = 23, //invalid firmware -> firmware upgrade is required
       AUTO_24LC_ADDR_ERR         = 24, //internal 24LC01 address (offset) is out of range
       AUTO_DM_ADDR_ERR           = 25, //DM (EEPROM) range error
       AUTO_FIRM_ERR              = 26, //firmware version error (old firmware)
       AUTO_NO_SUB                = 27, //no ICP-SUB PCB (obsolete)
       AUTO_NO_SUP_KEE            = 28, //no KeeLoq activation
       AUTO_NO_SUP_DSPIC          = 29, //no dsPIC/PIC24 activation (16-bit)
       AUTO_ICP2_REQ              = 30, //ICP2 required (ICP-01 programmer is not supported)
       AUTO_DEV_ERR               = 31, //device selection error (unspecified error)
       AUTO_PROG_MISMATCH         = 32, //mismatch between selected and detected programmers
       AUTO_PRJ_INVALID           = 33, //invalid environment
       AUTO_PRJ_DB_FIRM_PC_MIS    = 34, //mismatch between PC and firmware database
```

```
AUTO_PRJ_DB_FIRM_AT45_MIS    = 35, //mismatch between PJ2 and firmware database
AUTO_DLL_SUPPORT_REQIURED    = 36, //"GO" pressed on hardware and no DLL activation
AUTO_PRJ_CS                  = 37, //environment (PJ2) CS error
AUTO_STA_IDLE                = 38, //programmer is idle or standalone operation can't be started
AUTO_STA_BUSY                = 39, //standalone operation: programmer busy
AUTO_ENV_ERR                 = 40, //environment (PJ2) file error
AUTO_PM_RANGE                = 41, //invalid PM range specified
AUTO_SEC_SUPPORT_REQUIRED    = 42, //Security activation required
AUTO_SEC_CNT_INTEG           = 43, //Security feature: integrity error in counter
AUTO_SEC_CNT_ZERO            = 44, //Security feature: counter = 0
AUTO_SEC_NO_FUNC             = 45, //Security feature: function does not exist
AUTO_SEC_PACK_ERR            = 46, //Security feature: packet error
AUTO_SEC_EEPROM_FAIL         = 47, //Security feature: EEPROM error
AUTO_SEC_ANTI_SCAN           = 48, //Security feature: anti-scan activated
AUTO_SEC_SEC_ID_CMP          = 49, //Security feature: incorrect Security ID
AUTO_SEC_PASSW_CMP           = 50, //Security feature: incorrect password
AUTO_SEC_BATCH_CMP           = 51, //Security feature: incorrect batch
AUTO_SEC_VERS_ERR            = 52, //Security feature: version error
AUTO_SEC_UNKNOWN_ERR         = 53, //Security feature: unknown error
AUTO_NO_ROW_ERASE            = 54, //row/block erase is not supported
AUTO_INVALID_PARAM           = 55, //invalid parameters
AUTO_MOVLW_RETLW_CALIB       = 56, //no movlw in calibration word
AUTO_NO_USUAL_ENV_TRAN       = 57, //Usual environment can't be transferred if a secure one inside
AUTO_SEC_BUF_START_ADDR      = 58, //sec. buf. properties error: incorrect start addr
AUTO_SEC_BUF_END_ADDR        = 59, //sec. buf. properties error: incorrect end addr
AUTO_SEC_BUF_PAGE_START      = 60, //sec. buf. properties error: incorrect page start
AUTO_SEC_BUF_PAGE_SIZE       = 61, //sec. buf. properties error: incorrect page size
AUTO_SEC_BUF_NOT_EVEN        = 62, //sec. buf. properties error: length not even
AUTO_SEC_BUF_NO_DM           = 63, //sec. buf. properties error: no DM in PIC
AUTO_SEC_BUF_LAST_PAGE       = 64, //sec. buf. properties error: last PM page can't be used
AUTO_SEC_BUF_NO_16BIT_SUP    = 65, //sec. buf. properties error: no Script 1 for 16-bit devices
AUTO_SEC_BUF_NOT_MODULO_3    = 66, //sec. buf. properties error: length not modulo 3
AUTO_SEC_EMPTY_MASK          = 67, //Security feature: empty mask for secure environment
AUTO_TEST_COM_NO_SUPPORT     = 68, //ICP2 test command not supported
AUTO_TEST_NACK               = 69, //ICP2 test command returns NACK
AUTO_NO_SUP_P32              = 70, //no PIC32 activation
AUTO_PIC32_BUSY_OR_DAMAGED   = 71, //PIC32 is busy or damaged
AUTO_PIC32_CP_OR_DAMAGED     = 72, //PIC32 is code protected or damaged
AUTO_PIC32_PE_ANSWER         = 73, //PIC32 programming executive: no answer
AUTO_PIC32_PE_VERSION        = 74, //PIC32 programming executive: incorrect version (obsolete)
AUTO_SEC_BUF_NO_32BIT_SUP    = 75, //no security support for PIC32
AUTO_CNT_ZERO                = 76, //non-secure (low-endurance) counter is 0
AUTO_SQTP_CONFLICT           = 77, //serialization from PC is not allowed if standalone serialization=ON
AUTO_INVALID_DEVICE_CFG      = 78, //invalid device number in CFG file. Use latest DLL
AUTO_DEV_ID_NO_SUPPORT       = 79, //Device ID read is not supported for the family
AUTO_ROW_PM_RANGE            = 80, //invalid PM range due to row size
AUTO_PE_MISMATCH             = 81, //obsolete: Programming executive: mismatch between environment and firmware
AUTO_PE_NO_PGD_PULLDOWN      = 82, //No pull-down on PGD line (dsPIC33/PIC24)
AUTO_PE_VER                  = 83, //PE verification failed
AUTO_PE_NO_IN_ENV            = 84, //PE does not present in environment
AUTO_PE_CALIB                = 85, //invalid calibration/diagnostic data
AUTO_PC_DRV_STA_CONFLICT     = 86, //conflict between PC-driven and standalone modes
AUTO_CALIB_WORD_1_CORRUPT    = 87, //Calibration word 1 corrupted during programming
AUTO_CALIB_WORD_2_CORRUPT    = 88, //Calibration word 2 corrupted during programming
AUTO_ENV_NUM_OUT_RANGE       = 89, //Specified environment number is out of range
AUTO_CYBL_ACQUIRE_TIMEOUT    = 90, //Device acquire timeout or error
AUTO_CYBL_SROM_ACT_TIMEOUT   = 91, //SROM operation timeout
AUTO_CYBL_VIRGIN_DEVICE      = 92, //Device is VIRGIN
AUTO_CYBL_SWD_ACK_FAULT      = 93, //ACK response for SWD transfer is not OK (old name)
AUTO_SWD_ACK_FAULT           = 93, //ACK response for SWD transfer is not OK (new name)
AUTO_NO_FIRMWARE_CYBL        = 94, //no firmware for CYBL10x6x
AUTO_NO_FIRMWARE_I2C         = 95, //no firmware for I2C
AUTO_NO_FIRMWARE_DSPIC       = 96, //no firmware for dsPIC
AUTO_NO_FIRMWARE_P32         = 97, //no firmware for PIC32
AUTO_G3_REQUIRED             = 98, //G3 hardware is required for selected device
AUTO_G3_NO_PIC17C            = 99, //PIC17C is not supported by G3 programmers
AUTO_RESERVED_100            = 100, //reserved
AUTO_DEMO_ERR                = 101, //demo version
AUTO_OTP_NOT_BLANK           = 102, //OTP area is not blank, no programming is allowed
AUTO_OTP_VER_ERR             = 103, //OTP verification error
AUTO_FBOOT_VER_ERR           = 104, //FBOOT verification error
AUTO_DUAL_PART_ILLEGAL_BUF   = 105, //illegal partition mode in programming buffer
AUTO_DUAL_PART_MISMATCH      = 106, //partition mode mismatch
AUTO_NOT_ALLOWED_IN_DUAL     = 107, //operation is not allowed in dual partition mode
AUTO_DUAL_PART_ILLEGAL_PIC   = 108, //illegal partition mode in PIC
AUTO_FBOOT_BLANK_ERR         = 109, //FBOOT blank check error
AUTO_ENV_SIZE_ERR            = 110, //environment size is too big for connected programmer
AUTO_GANG_COMBO_MISMATCH     = 111, //mismatch between GANG and COMBO
AUTO_SWD_DEVICE_PROTECTED    = 112, //SWD device is protected
AUTO_DEVICE_PROTECTED        = 112, //Device is protected (same errorcode as for AUTO_SWD_DEVICE_PROTECTED)
AUTO_SECURITY_BIT_VER_ERR    = 113, //Security bit verification error
AUTO_CANT_CONNECT_TO_UUT     = 114, //can't connect to UUT
AUTO_SINGLE_WORD_RD_NO_SUP   = 115, //single word read not supported
AUTO_SINGLE_WORD_WR_NO_SUP   = 116, //single word write not supported
AUTO_ERASE_WRITE_TIMEOUT     = 117, //erase or write timeout
AUTO_UPDI_TINY_CRC_FAULT     = 118, //CRC fail. Execute programming with enabled bulk erase
AUTO_CANT_CONNECT_TO_UPDI    = 119, //can't connect to UPDI UUT (target)
AUTO_CANT_CONNECT_TO_TPI     = 120, //can't connect to TPI UUT (target)
AUTO_FTDI_LATENCY_BIG        = 121, //FTDI driver latency is too big
AUTO_FWU_CONS_INVALID        = 122, //Unexpected error: Cons.Valid is false
```

```
        AUTO_FWU_VERSION_3_2         = 123, //Firmware version must be 3.2 or greater
        AUTO_FWU_SELECTDEVICE        = 124, //Select Device error
        AUTO_FWU_NOT_FTB9            = 125, //Hex file error: not FTB9 firmware file
        AUTO_FWU_NOT_G3              = 126, //Hex file error: not G3 firmware file
        AUTO_FWU_NOT_ICP2            = 127, //Hex file error: not ICP2 firmware file
        AUTO_FWU_NOT_ICP01           = 128, //Hex file error: not ICP-01 firmware file
        AUTO_FWU_CHECKSUM            = 129, //Hex file checksum error
        AUTO_FWU_NOT_PORT_ACTIVATION = 130, //Selected HEX file is not a file for activation of ICP2-Portable options
        AUTO_FWU_NOT_ACTIVATION      = 131, //Selected HEX file is not a file for activation of options
        AUTO_FWU_VERSION_14_2        = 132, //Firmware 14.2 is not allowed
        AUTO_FWU_VERSION_14_0        = 133, //14.0 is the minimum firmware version for ICP2-Portable
        AUTO_FWU_INVALIDFILE         = 134, //Selected HEX file is not a file for firmware upgrade
        AUTO_FWU_VERSION_UNKNOWN     = 135, //Unknown firmware version
        AUTO_STM32L1_ERRATA_PCROP_NO_RDP = 136, //PCROP can't be enabled with RDP Level 0 - see errata sheet
        AUTO_REGION_0_NOT_BLANK      = 137, //Code region 0 is not blank.
                                            //See Preferences - Allow programming of pre-programmed device
        AUTO_ERASE_ERROR             = 138, //Erase error
        AUTO_PE_INTERNAL_ERR         = 139, //PE internal error (interaction between firmware and PE)
        AUTO_API_GEN_ERR             = 140, //API general error
        AUTO_API_FUSE_LOCKED         = 141, //CB (CMPA) is locked and can't be reprogrammed. Uncheck CB memory space
                                            //for flash programming
        AUTO_API_ADDR_ERR            = 142, //Address error in API
        AUTO_API_COMMAND_ERR         = 143, //Command error in API
        AUTO_DEBUG_RESTRICT          = 144, //Restricted debug access
        AUTO_ECC_MISMATCH            = 145, //ECC settings in buffer and target are different. Can not proceed.
        AUTO_UHFL_INVALID_PARAM      = 146, //Universal Hex File Loader: invalid conversion string
        AUTO_CANT_CONNECT_TO_EXTERN_POWERED_UUT = 147, //can't connect to externally powered UUT
        AUTO_DEVICE_ID_ERR           = 148, //Incorrect Device ID
        AUTO_SWAP_SET_NOT_ALLOWED    = 149, //SWAP setting is not allowed (not supported on STM32H)
        AUTO_DEBUG_ERR               = 150, //Debug error (internal R&D error)
        AUTO_INIT_ERR                = 151, //Initialization error (during complicated connection to target)
        AUTO_NO_FAM_SUPPORTED        = 152, //Family of selected device is not activated, contact Softlog
        AUTO_FAM_SUPPORT_FIRM        = 153, //Firmware 34.2 or higher is required for FAM (family) activation system
        AUTO_MAILBOX_INVALID_NUM     = 154, //Mailbox invalid number
        AUTO_MAILBOX_INVALID_VALUE   = 155, //Invalid value in mailbox
        AUTO_MAILBOX_CS              = 156, //Mailbox checksum error
        AUTO_RAM_BUF_CANT_BE_USED    = 157, //RAM buffer can't be used
        AUTO_CALIB_ERR               = 158, //Calibration error
        AUTO_CALIB_BAD_VALUE         = 159, //Calibration value out of range
        AUTO_CALIB_LEN               = 160, //Calibration value length out of range
        AUTO_CALIB_ALG               = 161, //Calibration algorithm number out of range
        AUTO_CALIB_FREQ              = 162, //Calibration frequency out of range
        AUTO_CALIB_TO_BAD_VALUE      = 163, //Calibration timeout out of range
        AUTO_CALIB_TMP_AREA          = 164, //Invalid area for temporary storage
        AUTO_CALIB_TMP_OFFSET        = 165, //Invalid offset for temporary storage
        AUTO_CALIB_FINAL_AREA        = 166, //Invalid area for final storage
        AUTO_CALIB_FINAL_OFFSET      = 167, //Invalid offset for final storage
};
```

33 Appendix C: DEBUG and COE Bits

A compiler may provide incorrect settings for debug-related bits: DEBUG (Background debug) and/or COE (Clip-on emulation mode). Development tool as MPLAB® IDE manipulate these bits automatically while ICP family software loads them from a HEX file "as is".

The following procedure is strongly recommended to validate these bits:

- Run MPLAB IDE
- Compile in "Release" mode
- Export the HEX file (File → Export) under a name (for example) "1.hex"
- Import HEX file "1.hex" (File → Import")
- Write down the checksum

- Run "ICP for Windows"
- Open "1.hex"
- Compare checksums
- Inspect configuration bits. If DEBUG or COE are set to debug mode then change them to operational one
- Save updated buffers: File → Save As (Export)…

34 Appendix D: System Debug Procedure

- Validate that your programming cable is OK
  - Clock (pin 3) and Data (pin 4) are not swapped
  - VPP/MCLR comes from pin 6
  - If possible then use separate short wires (less than 20cm during debug)

- Debug the system in full PC-driven mode which provides much more info
  - Switch the Control Center to PC-driven mode
  - For ICP2-GANG/COMBO only: enter Programmer → Select Programmer → ICP2-GANG/COMBO (**Single**)
  - For ICP2-GANG/COMBO only: on the popup window select desired channel
  - Connect your target to the selected TARGET connector
  - Validate that the correct PIC is selected
  - Validate that Voltages and Preferences settings are OK
  - Load your HEX
  - Execute programming

- If the problem persists then send the following info to support@softlog.com:
  - Screenshot of entire screen including status line
  - Verification error screen
  - Electrical circuit, at least connections to VDD/MCLR/CLOCK/DATA
  - Description of the programming cable:
    - length (should be as short as possible)
    - cable type (separate unshielded wires are recommended)

35 Appendix E: Recommendations for Low Pin Count Devices

Follow recommendations below in your firmware for low pin count devices in order to allow correct entering to programming mode:
- enable power-up timer
- enable MCLR/VPP pin if possible
- add delay 50-200ms before configuring shared programming pins as outputs

36 Appendix F: Emergency Bootloader Mode

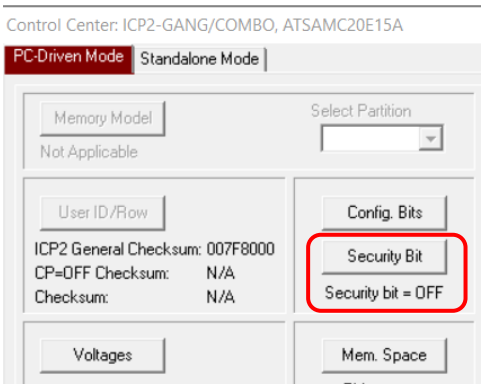
The following sequence can be executed in order to enable "emergency bootloader" mode which temporarily disables transition to the firmware

- Preparation: connect a push button (let's call it START) between pins 12 (GND) and 13 (GO) of the TARGET connector
- Preparation (CP210x only): disable power management on the USB port:
  - exit "ICP for Windows"
  - connect ICP family programmer to USB port
  - enter "Control Panel - System - Device Manager - Ports(COM &LPT) - select "Silicon Labs 210x USB…"
  - press right mouse button and select "Properties"
  - select TAB "Power Management"
  - uncheck "Allow the computer to turn off this device to save power"
  - press OK
- Disconnect ICP2 from USB and power supply (power should be off)
- Press and hold START button
- Connect ICP2 to the USB port, START button should be kept pressed
- When red LED "FAIL" goes on, immediately release START button
- Green LED "PASS" starts to blink. It means that ICP2 entered the bootloader mode
- Run "ICP for Windows"
- If no communication: enter "Connect" and select "Silicon Labs 210x USB…". Press OK
- Execute firmware upgrade when prompted or select Options - Firmware Upgrade

37 Appendix I: Special Code Protection Settings

37.1 Security Bit

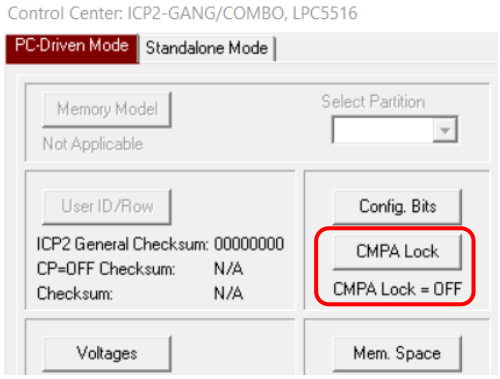
- Security bit is available on devices like ATSAMC/D/L/R, for example ATSAMC20E15A
- It can be set as follows in order to enable code protection:
  - ICP for Windows: "Security Bit" on the Control Center



- ICP DLL: function IcpLoadSecurityBit()
- ICP Command Line: switch /y

37.2 CMPA Lock

- CMPA Lock bit is available on devices like LPC55xx, for example LPC5516
- It can be set as follows:
  - ICP for Windows: "CMPA Lock" on the Control Center



- ICP DLL: function IcpLoadSecurityBit()
- ICP Command Line: switch /y

38 Appendix J: Oscillator Frequency Calibration

- PC-driven and fully standalone oscillator frequency calibration is available for several Atmel SPI devices (ATmega328P, etc.). See document:
  c:\Softlog\IcpWin\Release Notes\Release Notes - ATmega calibration.pdf

39 Revision History

- Revision 12.1.3a (Jan-22):
  - updated connection table
  - "Control Interface" for ICP2-COMBO: added truth table for environment selection
  - added paragraph "Emergency Bootloader Mode"
  - added paragraph "Special Code Protection Settings"
  - added paragraph "Oscillator Frequency Calibration"
  - minor updates

- Revision 8.35.1a (Jan-20):
  - updated connection table in paragraph
  - removed references to ICP-01 (obsolete product since year 2007)
  - added new menu items, see paragraph 12
  - expanded ICP2-GANG and ICP2-COMBO: Transfer Different Environments to Different Channels
  - added PC-Driven Serialization in Standalone Mode
  - added new Preferences items – see Options → Preferences

- added permanent "Gap Eliminator Summary" window – see 20.1
- added RESET Pulse after Operation
- added Appendix E: Recommendations for Low Pin Count Devices
- added recommended power supply part numbers – see 31
- removed paragraph "Advanced LAN Configuration"

- Revision 8.8.1a (Jul-18):
  - added reference to ICP2-Portable(G3) programmer
  - added reference to devices with new interfaces (UPDI, TPI, etc.)
  - added new Preferences

- Revision 8.1.1a (Mar-17):
  - added reference to G3 programmers: ICP2(G3), ICP2-GANG(G3) and ICP2-COMBO(G3)

- Revision 4.16.1a (Jan-16):
  - added maximum COM port warning
  - added description of manual environment procedure (preferences and serialization, different environment for different channel)
  - added system debug procedure

- Revision 4.13.1a (Jan-15):
  - changed ICP software setup destination (new: C\Softlog\..., old: C:\Program Files\Soft-Log)
  - added ICP2-COMBO related info
  - added checksum calculation explanation
  - added LAN description

- Revision 4.12.1 (Aug-13):
  - added warning for serialization dialog – see 16.1
  - corrected example 16.2
  - added warning for ICP2-Portable

- Revision 4.10.2 (Aug-12):
  - added "Speed Optimization Utility" paragraph
  - added "Gap Eliminator™" paragraph
  - added new Preferences

- Revision 4.9.2 (Apr-2012):
  - added description of Enhanced ICSP limitations – see 19.6
  - changed appearance of "Preferences"

- Revision 4.9.1 (Jan-2012):
  - added description of "Enhanced ICSP™"
  - GO/PASS/FAIL outputs specified as optional for ICP2-Portable

- Revision 4.8.2 (Aug-2011):
  - added description of "raw" serialization method, serialization screenshots updated

- Revision 4.8.1 (Jul-2011):
  - added Preference "Automatically select PM (flash) range"
  - added standalone serialization – see 16.6
  - added paragraph "Viewing Environment" - see 15
  - added paragraph "Debug and COE Bits" - see 33

40 Technical Assistance

You may contact Softlog Systems for technical assistance by calling, sending a fax or e-mail. To help us give you quick and accurate assistance, please provide the following information:

- Software version number, firmware version number and product serial number (if available). This information is displayed at the program start
- Detailed description of the problem you are experiencing

- Error messages (if any)
- Microcontroller part number (if device-related)
- Send us your "icp01.cfg" file

41 Warranty

Softlog Systems (2006) Ltd. warrants this product against defects in materials and workmanship for a period of 1 (one) year. This warranty will not cover programmers that, in the opinion of Softlog Systems, have been damaged due to abuse, improper use, disassembly, replacement of parts or attempted repair by anyone other than an authorized Softlog Systems service technician.

This product must be returned to the supplier for warranty service within the stated period. The buyer shall pay all shipping costs and other charges or assessments for the product by the supplier.

Softlog Systems shall not be liable for any indirect, incidental, or consequential damages, regardless of whether liability is based upon breach of warranty, negligence, strict liability in tort, or any other theory, Softlog Systems will never be liable in an amount greater than the purchase price of the products described by this express warranty. No agent, distributor, salesperson, or wholesale or retail dealer has the authority to bind Softlog Systems to any other affirmation, representation, or warranty concerning these goods.

42 Contact

Softlog Systems (2006) Ltd.

6 Hayotzrim St. Or-Yehuda 6021820 Israel

Phone: 972-3-9515359
Fax: 972-3-9527520
Web: www.softlog.com
E-mail: sales@softlog.com, support@softlog.com

43 Copyright Notice