**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"
- ICP2-ISO(G3) and ICP2-LAN(G3) products are also referred as ICP2-ISO and ICP2-LAN

# 1   Connectors

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

| Pin No. | Pin Name (ICP2-GANG, ICP2-COMBO, ICP2-ISO, ICP2-LAN) | Pin Name (ICP2, 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 and ICP2-ISO

## 1.3   "LAN" Connector: Standard RJ-45

Note: available on ICP2-COMBO and ICP2-LAN

## 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<br>**ICP2**: additional power supply input |
| 5 | **GND** | - | GND | Ground connection |
| 6 | 12V_OUT | +11…14V | Power | **ICP2-GANG**: power output<br>**ICP2/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 |
| B4 | CHAIN_232_TXD_ISO | Standard RS-232 output | RS-232 output to chained ICP2-COMBO |
| 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: |
| 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 | (see table below) |

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

| Pin No. | Pin Name | Pin Type | Description |
|---|---|---|---|
| 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-Portable/ICP2-ISO/ICP2-LAN      - 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

# ICP Family User's Manual

## 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 "RS-232, CONTROL" Connector (Header-16 Male, ICP2-ISO: 2.54mm, ICP2-LAN: 2mm)
NOTE: all voltages are referred to **isolated** GND (GND_ISO)

**ICP2-ISO mating connector:** HIF3FB-16DA-2.54DSA(71) (Hirose), header 16 pin, 2.54mm, female, with key (polarized)

**ICP2-LAN mating connector:** 79107-7007 (Molex), header 16 pin, 2mm, female

| Pin No. | Pin Name | Pin Type | Description |
|---|---|---|---|
| 1 | PC_RXD_ISO | Standard RS-232 output | RS-232 output to PC |
| 2 | GND_ISO | - | Isolated GND |
| 3 | PC_TXD_ISO | Standard RS-232 input | RS-232 input from PC |
| 4 | GND_ISO | - | Isolated GND |
| 5 | CHAIN_232_TXD_ISO | Standard RS-232 output | RS-232 output to chained ICP2-ISO |
| 6 | GND_ISO | - | Isolated GND |
| 7 | CHAIN_232_RXD_ISO | Standard RS-232 input | RS-232 input from chained ICP2-ISO |

| Pin No. | Pin Name | Pin Type | Description |
|---------|----------|----------|-------------|
| 8 | GND_ISO | - | Isolated GND |
| 9 | GO_ISO | CMOS input (0/Z) with weak pull-up | Programming activation (active low) |
| 10 | PASS_ISO | CMOS output with in-series R=330Ω | Pass/busy indication |
| 11 | FAIL_ISO | CMOS output with in-series R=330Ω | Fail/busy indication |
| 12 | 5V_ISO | 5V reference source | Isolated 5V with in-series fuse 50mA, can be used to control ENV_SEL_A/B/C_ISO lines |
| 13 | ENV_SEL_A_ISO | CMOS input (0/5V) with weak pull-down | Environment selection: |
| 14 | ENV_SEL_B_ISO | CMOS input (0/5V) with weak pull-down | |
| 15 | ENV_SEL_C_ISO | CMOS input (0/5V) with weak pull-down | |
| 16 | Reserved | CMOS input (0/5V) with weak pull-down | Future use |

Environment selection (Pin 13, 14, 15):

| …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

### 1.10   Typical Connection to "TARGET" Connector

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

## 2   PC-Driven and Standalone Modes

### 2.1   Modes

ICP family programmers can be operated in full PC-driven, standalone mode (PC-controlled) and full standalone mode (w/o PC)

| Programmer | Full PC-Driven | Standalone (PC-controlled) | Full Standalone (w/o PC) |
|------------|----------------|----------------------------|--------------------------|
| ICP2 | Yes | Yes | Yes |
| ICP2-GANG, single channel | Yes | Yes | Yes |
| ICP2-GANG, multi-channel | - | Yes | Yes |
| ICP2-COMBO, single channel | Yes | Yes | Yes |
| ICP2-COMBO, multi-channel | - | Yes | Yes |
| ICP2-Portable | Yes | Yes | Yes |
| ICP2-ISO, single channel | Yes | Yes | Yes |
| ICP2-LAN, single channel | Yes | Yes | Yes |
| ICP2-ISO, multi-channel | - | Yes | Yes |
| ICP2-LAN, multi-channel | - | 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.).a

*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"

### *2.2 Standalone Programming*

True parallel multi-channel programming can be done in standalone mode only. The standalone programming can be done in 3 ways:

- From PC:
  - ICP for Windows
  - ICP DLL
  - ICP Command Line / Advanced Command Line
  - ICP Secure Programming
- GO input on the programmer unit (optional on ICP2-Portable) by GO/PASS/FAIL wired interface
- RS-232 high-level communication (DLL/Command Line support should be activated)

# 3 Chain Connection (ICP2-GANG, ICP2-COMBO, ICP2-ISO and ICP2-LAN)

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

Number of channels can be increased up to 16 by daisy chain connection between ICP2-ISO or/and ICP2-LAN units (they can be mixed)

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 (see Notes) | | 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:<br>- Vdd overload<br>- Vpp overload<br>- I2C communication error | OFF | Blink | OFF | ON |
| 6. | Non-UUT problem during standalone operation:<br>- database error<br>- device not supported<br>- no family activation<br>- 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) | |

Notes:
- ICP2-ISO and ICP2-LAN PASS/FAIL LEDs provide additional blinks on power-up as follows:
  - 1 blink: configured as "Channel ≠ 2" (i.e. non-default channel 2)
  - 2 blinks: configured as "Use RS-232 interface only"

## 5 Other LEDs (ICP2-Portable)

| ## | LED Name | Color | LED Behavior |
|---|---|---|---|
| 1. | POWER | **Blue** | ON |
| 2. | LOW BATTERY | **Red** | • Battery is normal: OFF<br>• Battery is low: ON<br>• "GO" button is pressed & battery is very low:<br>no programming is allowed, blinks in parallel to FAIL LED |
| 3. | SERIALIZATION | **Green** | • Serialization = OFF: OFF<br>• Serialization = ON & S/N (serial numbers) = OK: ON<br>• Serialization = ON & no S/N: permanent slow blink<br>• 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<br>• Counter = 0: permanent slow blink<br>• Counter = 0 & "GO" button is pressed: no programming is allowed, blinks in parallel to FAIL LED |
| 5. | Environment<br>(1 of 6 LEDs) | **Green** | • Environment is selected & OK: ON<br>• Environment is selected & invalid: permanent slow blink<br>• 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

### 6.1 Standalone Programming using GO/PASS/FAIL Interface

- Prepare environments and transfer to programmer – see paragraph 14 for details
  NOTE: programmer is automatically ready for programming once the environments are 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

### 6.2 Standalone Programming using High-Level RS-232 Communication

- Prepare environments and transfer to programmer – see paragraph 14 for details
  NOTE: programmer is automatically ready for programming once the environments are saved in **non-volatile** memory
- Follow document *"High-Level RS-232 Communication.pdf"*

## 7 Host Computer Requirements

- 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
- Free USB port (all programmers excluding ICP2-LAN), free RS-232 port (all programmers excluding ICP2-Portable) or free LAN connection (ICP2-COMBO or ICP2-LAN)

## 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: https://softlog.com/downloads/

#### 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, ICP2 and ICP2-ISO)
- 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 is placed into a closed equipment

### 8.5 ICP2-Portable Setup

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

### 8.6 ICP2 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)** and ICP2-ISO(G3) programmers operate 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).
   Notes:
   - 1ms is not recommended due to possible communication problems on several computers
   - set to 4ms is communication problems arise with 2ms (rare cases)
  - 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: sales@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 by default (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 or marking "Clear all memory buffers before loading" in "Open (Import)" dialog

## 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 data file from disk and load it into buffer memory area |
| Save | Save the currently loaded file in HEX format |
| Save As (Export)... | Save the buffer to a HEX file on disk |
| Universal HEX Loader | Load a data file with virtual addresses (usually ATmega/ATtiny/AVR) |
| Load PM with zero offset | Load a data file with zero offset into PM (flash) |
| Load BM with zero offset | Load a data file with zero offset into BM (boot memory) |
| Import Data Memory | Load a data 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 Multi-Channel Programmer | Transfer different PJ2 files to programmer in single dialog |
| Allow Sub-Environments | Turn on/off sub-environment mechanism |
| 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 ICP2-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 |
| ICP2-ISO/LAN Configuration | Select active ICP2-ISO/LAN channels (16 max) |
| Assign Address and Baud Rate to ICP2-ISO/LAN Unit | Assign channel address to currently connected programmer (1-16) |
| 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 |
| Close COM | Close COM |

### 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 |
| Create Device/Family Preliminary List for Activation (offline) | Future use |
| Summary of Options/Families, Collect My Firmware IDs | View summary of options/families, save Firmware IDs for activation |
| Activate Options/Families by HEX File | 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** |
| OEM Solutions | Enable OEM features, **contact Softlog Systems for 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 |
| <Different documents> | |
| About | Connect to the programmer and display software and firmware versions, programmer details and activated families |

## 13  Shortcuts

| | |
|---|---|
| Save | Ctrl-S |
| Open (Import)... | Ctrl-O |
| Program | F5 |
| Verify | F6 |
| Blank Check | F7 |
| Read | F8 |
| Programming Only | F9 |
| Programming buffers: go to address | Ctrl-G |

# 14 Preparing and Transferring Environment to Programmer

## 14.1 Sub-Environment Mechanism

- Starting from software 16.1.1(beta) 9-Feb-25 till 512 environments can be used.
  See "*Release Notes: Sub-Environments.pdf*" for details (it's also available by pressing "Help" on the environment "Destination" window – see a screenshot below)
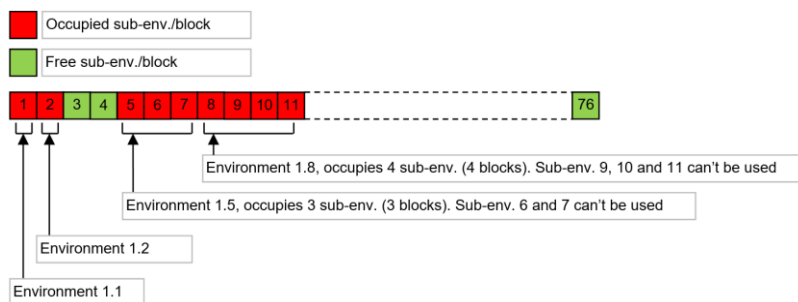


## 14.2 Prepare and Transfer Environment with Wizard

- Run "Environment/Environment Wizard" and follow the Wizard
- Select programmer and press "Next"
- ICP2-GANG/ICP2-COMBO: 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 can 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

## 14.3 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
- Set 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…

### 14.4 ICP2-GANG and ICP2-COMBO: Transfer Different Environments to Different Channels

14.4.1   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 14.3 (don't transfer)

- Select Environment → Transfer Different Environments to ICP2-GANG/ICP2-COMBO and load PJ2 files (Env5.pj2…Env11.pj2 are not shown)
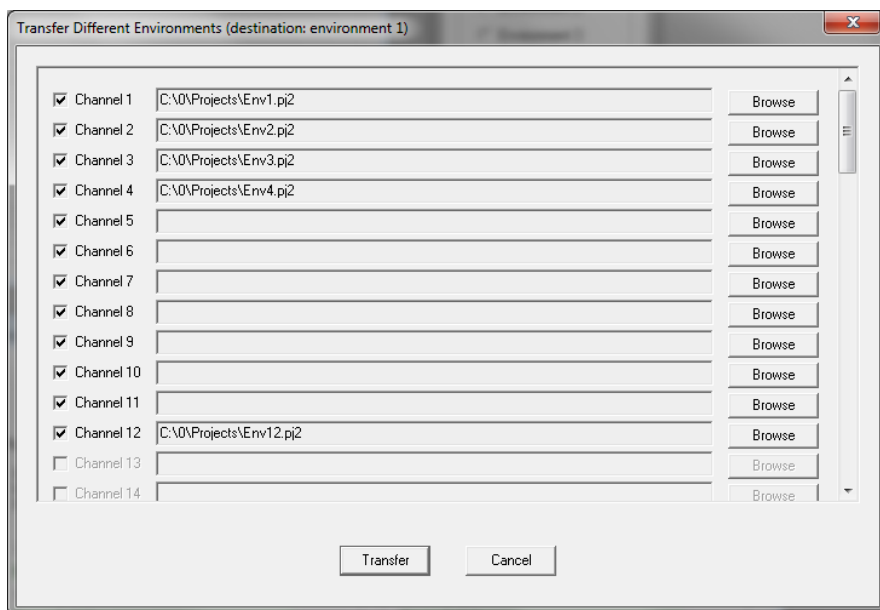


- Press "Transfer"

14.4.2   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 14.3 (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
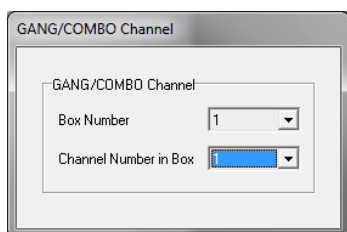


- Transfer Env1.pj2: Environment → Transfer Environment to Programmer…

- 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
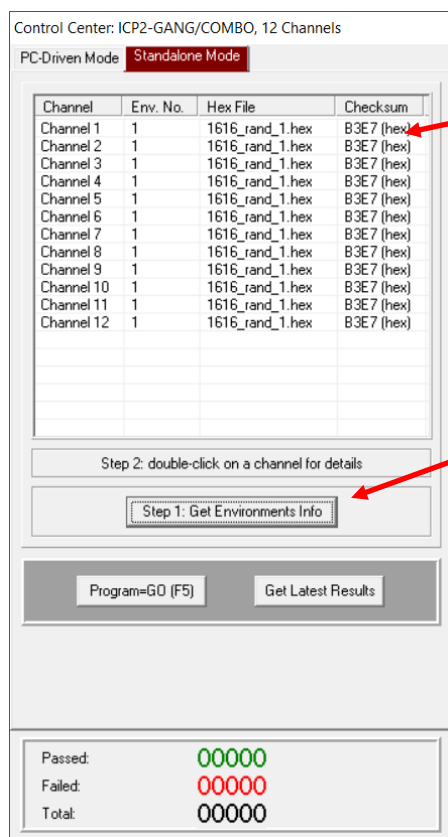


Step 2: Double-click

Step 1: Click once

Step 3: Select desired TAB
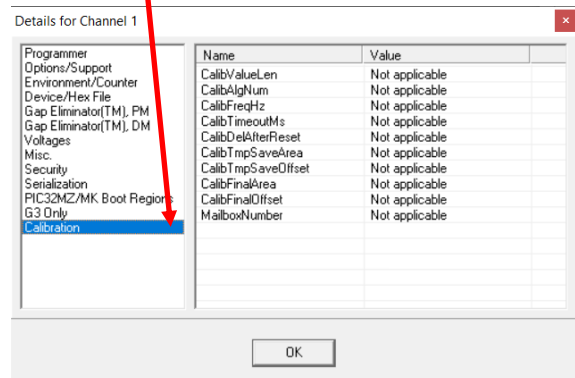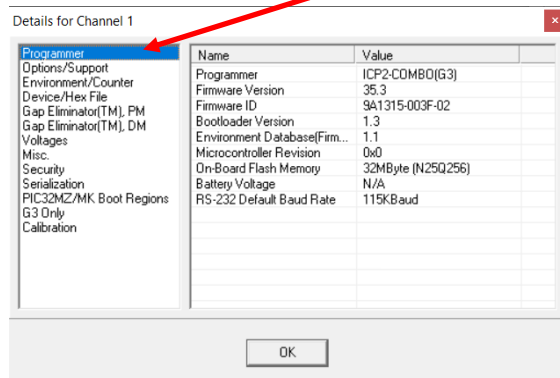
# 16  Serialization

## 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

  <mark>WARNING:</mark> 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)

**Chip Programming (Standalone Mode)**

Standalone operation in progress channel 12 (box 3.4)

| Channel | Status | Serial Number (PC) | Result | Verification Errors |
|---------|--------|--------------------|--------|---------------------|
| Channel 1 | Programming... | 123462 | Busy | |
| Channel 2 | Programming... | 123463 | Busy | |
| Channel 3 | Programming... | 123464 | Busy | |
| Channel 4 | Programming... | 123465 | Busy | |
| Channel 5 | Programming... | 123466 | Busy | |
| Channel 6 | Programming... | 123467 | Busy | |
| Channel 7 | Programming | 123468 | Busy | |

Cancel

# 17 Options → Voltages

**Set Voltage**

Vdd Source
- ⦿ Programmer
- ◯ Target

VddProg Mode
- ◯ Database
- ⦿ As VddMax

I/O Voltage
- ⦿ As Vdd
- ◯ From this dialog

Voltages

| | |
|---|---|
| VddMin, V: | 3.30 |
| VddMax, V: | 3.30 |
| Vpp, V: | 11.50 |
| Vio, V | As Vdd |
| Vdd-to-Vpp, ms: | 4 |

Advanced   OK   Cancel

### 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 before/during 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   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

### 19.8   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

### 19.9   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

### 19.10  Smart hard. erase/unlock, factory reset [ ]

Applicable to devices with hardware erase pin (ATSAM3/4/7, etc.) or special unlock or factory reset (MSPM0G/L, TM4C). If the option is ON then related operation is executed

### 19.11  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)

### 19.12  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™

### 19.13  Atmel SPI / CYBLE / ESP32 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, faster erase/programming mechanism for ESP32

### 19.14 Calculate fuse CRC or CS of inter. vectors [ ]

Applicable to several devices only

### 19.15 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

### 19.16 Allow programming of pre-programmed device [ ]

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

### 19.17 Connect under reset [ ]

Applicable to several devices

### 19.18 Check Device ID [x]

Checks Device ID before programming

### 19.19 Force DM (EEPROM) erase [ ]

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

### 19.20 Alternative access port [ ]

Applicable to several devices

### 19.21 Use calibration value from mailbox [ ]

Applicable to ATmega devices when frequency calibration is required

### 19.22 CRC/MD5 verification [ ]

Applicable to devices with active flashloader for faster verification

### 19.23 Swap clock/data pins (SBW) [ ]

Swap clock and data pins for devices with SBW interface

### 19.24 Power cycle entry
  *(●)  Auto*
  *( )  Vdd ≤ 3.6 V*
  *( )  Vdd > 3.6V*

Applicable to CY8C2 devices

### 19.25 Don't program protected devices [ ]

Don't program protected MCUs to avoid decrement of the non-volatile secure counter

### 19.26 Reduce synchronization frequency [ ]

Reduce operation frequency by additional step after synchronization is done, applicable to Atmel MCUs with SPI interface (ATmega, ATtiny)

# 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 for all Softlog's programmers.

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



Click on "Range" or "Gap" for fast navigation in PM or DM buffer
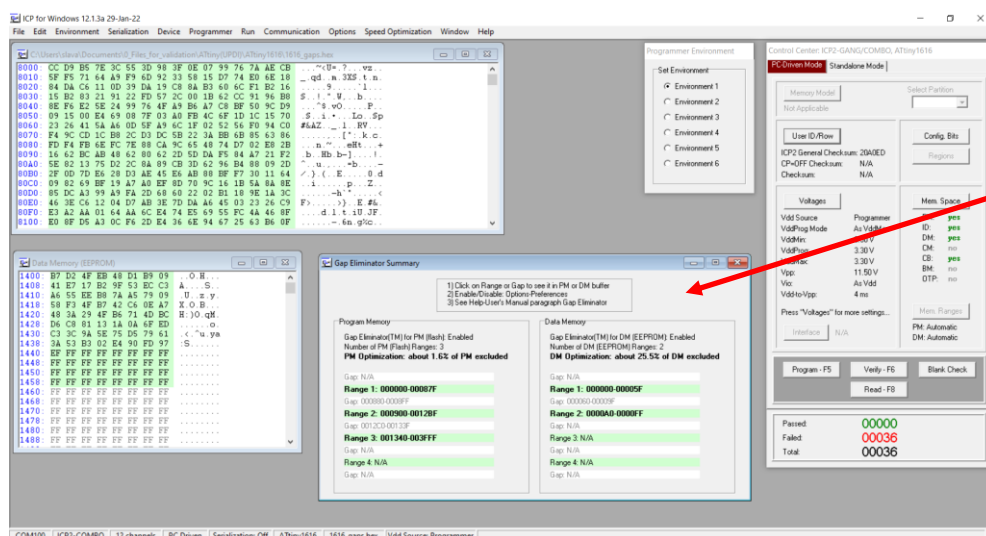
## 20.2 How It Works

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 **<u>not</u>** 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/BIN/LDF 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
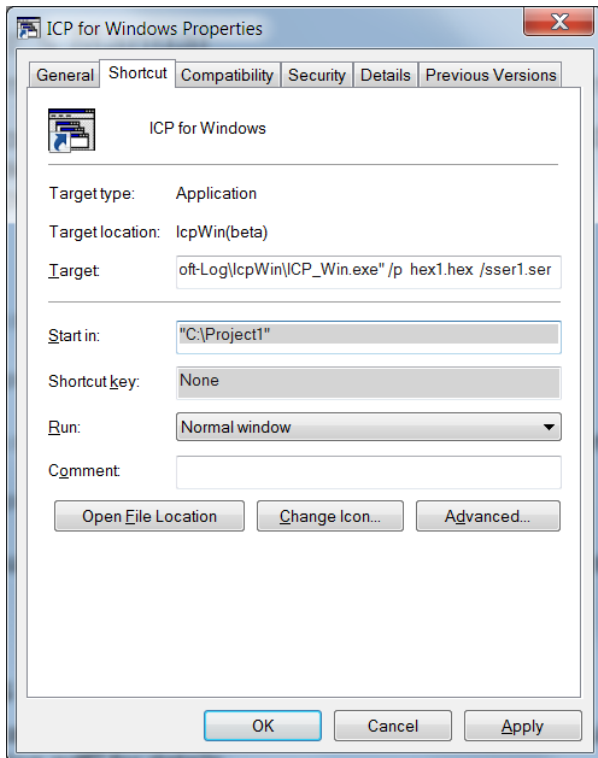
## 24 DLL Functions

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

NOTES:
- See *"ICP LAN Configurator User's Manual.pdf"* for detailed description of all features
- Screenshots below are taken from the previous software version but they completely cover the latest version

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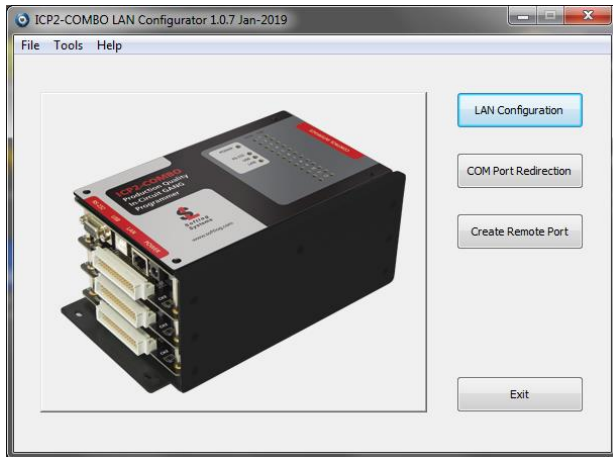
The "ICP-LAN" application provides 3 configuration features:
- "LAN Configuration" which discovers ICP2-COMBO or ICP2-LAN and changes IP network parameters
- "COM Port Redirection" which creates a virtual COM (CPR) port for ICP2-COMBO or ICP2-LAN
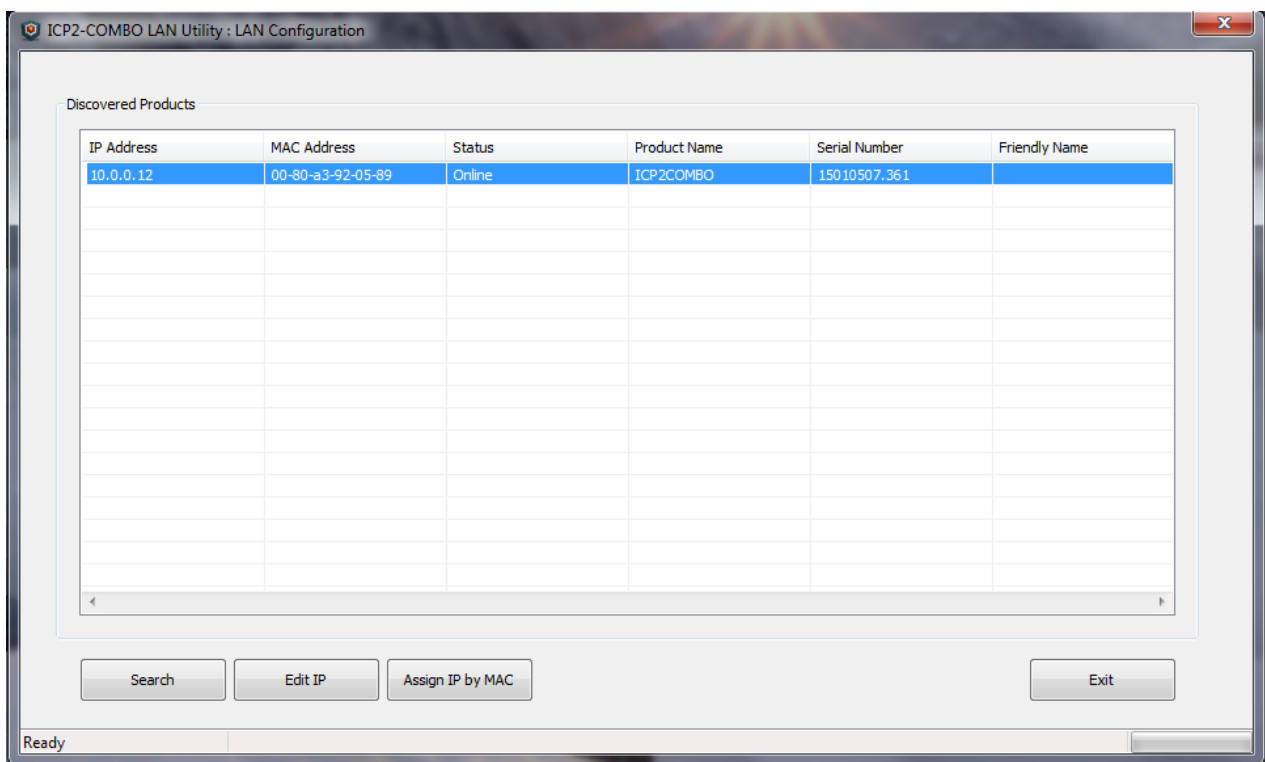- "Create Remote Port" which creates remote port w/o discover procedure

## 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 or ICP2-LAN sticker
- if your device was not found refer to "Advanced IP Configuration"
- 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:
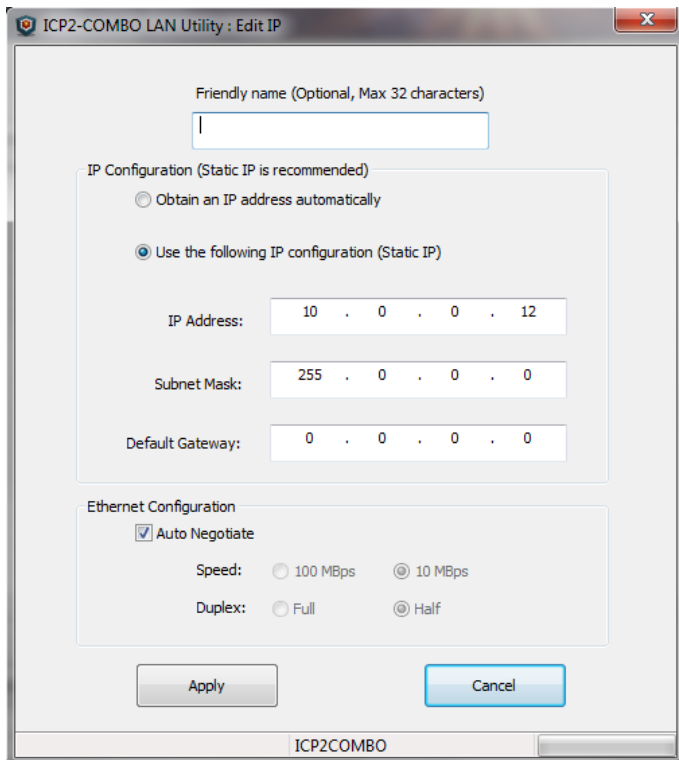
- Change your settings if required

  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:
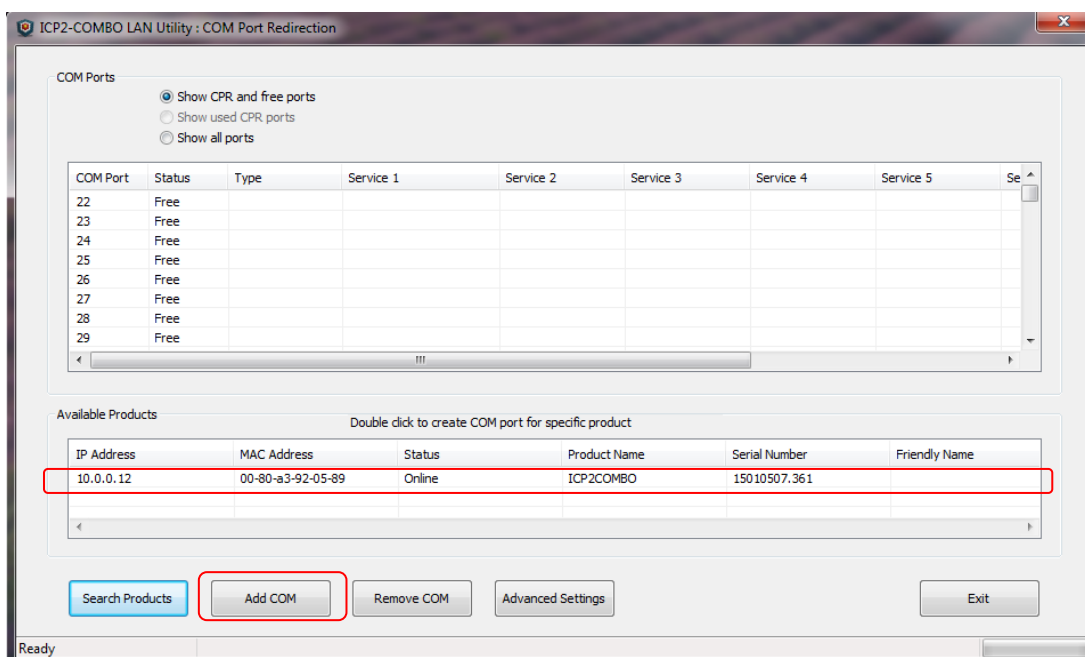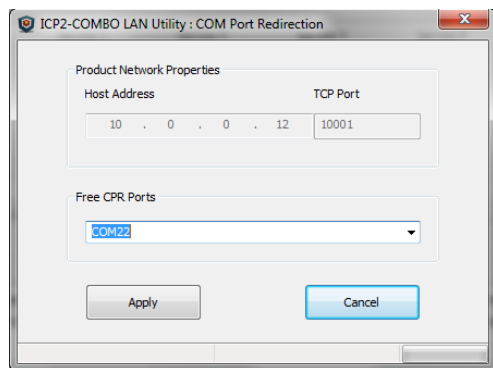


- Select desired COM port number and press "Apply". COM Ports list on "COM Port Redirection" window will refresh automatically:
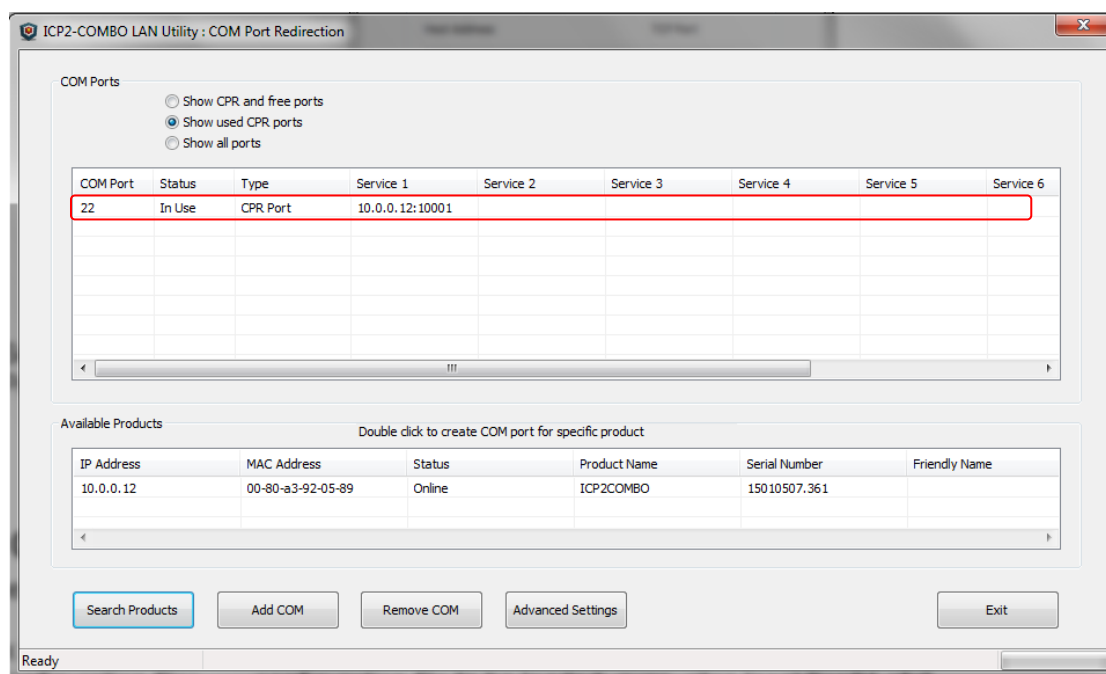


- Press "Exit"

## 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 related programmer specification:
- "ICP2(G3) Specification.pdf"
- "ICP2-GANG(G3) Specification.pdf"
- "ICP2-COMBO(G3) Specification.pdf"
- "ICP2-Portable(G3) Specification.pdf"
- "ICP2-ISO(G3) Specification.pdf"
- "ICP2-LAN(G3) Specification.pdf"

## 29.1 VDD

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

- Maximum Vdd capacitance: 1000-10000uF. 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: 18-33pF between CLOCK and GND **as close as possible** to the target microcontroller is recommended, especially for long cables (> 20cm).
- See *"Filter for long cables.pdf"* in folder "c:\Softlog\IcpWin\Misc Documents\"

# 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-ISO | 12VDC | 0.5A | "+" | |
| ICP2-LAN | 12VDC | 0.5A | "+" | |
| ICP2-Portable | 12VDC | 0.5A | "-" or "+" | |
| ICP2-GANG | 12VDC | 1.5A | "+" | |
| ICP2-COMBO | 12VDC | 5A | "+" | GST60A12-P1J (Mean Well) |

## 32 Appendix B: Return Values (Errorcodes)

See file *"fr_exp.h"* or ICP for Windows – Help – Errorcodes (fr_exp)

## 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 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 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 to enable code protection:
  - ICP for Windows: "Security Bit" on the Control Center
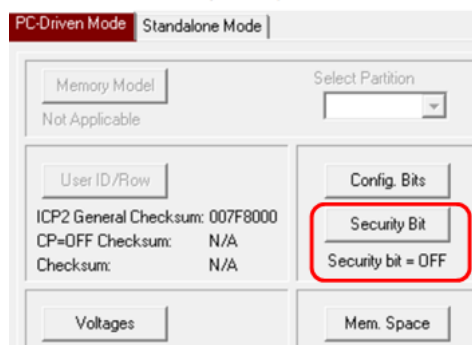


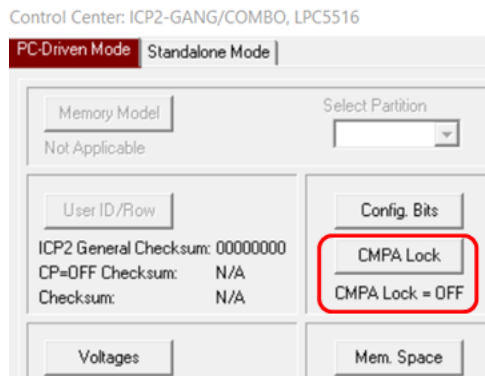  - ICP DLL: function IcpLoadSecurityBit()
  - ICP Command Line: switch /y

### 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 Mar-25:
  - added fully standalone mode with high-level RS-232 communication
  - added sub-environment mechanism

- Revision Mar-24:
  - added ICP2-ISO(G3) and ICP2-LAN(G3) programmers
  - removed reference to ICP2(HC)
  - added reference to BIN and S19/S28/S37 data files
  - removed "Typical Connection to TARGET Connector": now it's referred to a separate document
  - removed "Errorcodes": now it's referred to a separate document

- 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

Windows is a registered trademark of Microsoft Corporation. Microchip, MPLAB, PIC, dsPIC and ICSP are registered trademarks of Microchip Technology Incorporated.