

Introduction

The **ProISP IV** from **PROCHILD** is a low cost In-System Programmer covering all **AVR MCUs** with **ISP, PDI** or **TPI** Interface.

<Features>

- Stand-alone **AVR** Programmer(No PC needs)
- Easy to use **GUI**
- Programs both **Flash** and **EEPROM**
- Supports **Fuse** and **Lock** byte programming
- Supports **RC Oscillator Calibration byte** programming
- Supports **Serializing**
- Upgradable for future devices
- **USB** Interface

<Contents>



ProISP IV

USB cable

PROCHILD ProISP IV

Getting Started

1. Install 'ProISP IV PROGRAMMER' and USB driver

- **USB driver**

Before connecting ProISP IV to your PC you have to install a USB to UART Bridge VCP(CP2102) driver

- **Software**

In order to use the **ProISP IV** it is required to install the **ProISP IV PROGRAMMER**. You can download on our website(see below)

<Download>

- <http://www.prochild.co.kr> -> DATA room
 - > S/W : ProISP IV PROGRAMMER Version x.xx
 - > Driver : [Silicon Labs]CP210x USB to UART Bridge VCP Driver
(It can be downloaded from <http://www.silabs.com>)

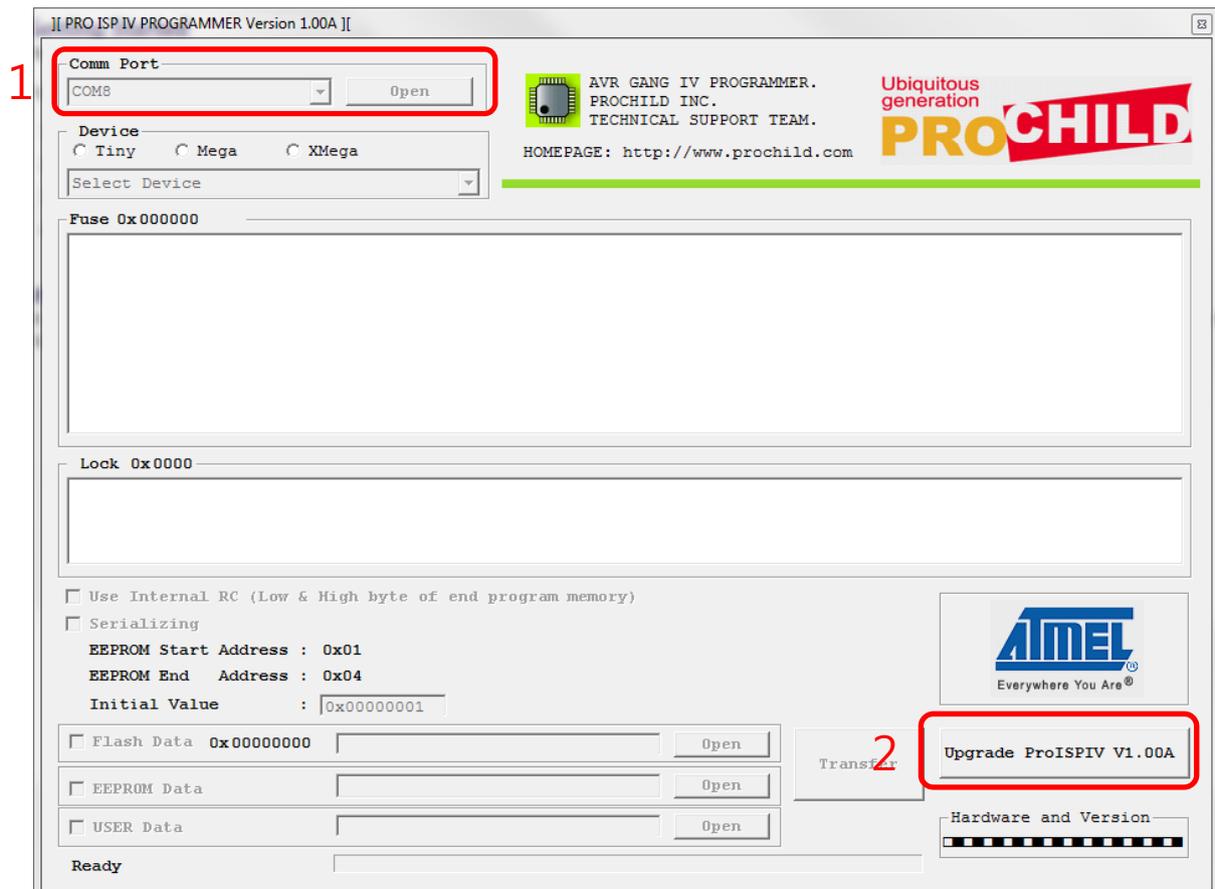
2. Mode Switch

- Set the **Mode Switch** to '**U**' when you upgrade ProISP IV or update source(.hex) code
- After update code, you should set the **Mode Switch** to '**T**' for target writing

Getting Started(Cont.)

3. Upgrade ProISP IV

- Start '**ProISP IV PRGRAMMER**'.
- Connect **ProISP IV** to PC using USB cable.
- Set the **Mode Switch** to '**U**'
- Open a correct **Com Port**.
(Check it on the *Device Manger* provided by windows.)



Download & Target Writing

1. To program your target devices, you have to download your code to the **ProISP IV** first.
 - Open **Com Port** → Select **Device** → Set **Fuse** → Set **Lock** → Open a code(.Hex) → **Transfer**

The screenshot shows the PROISP IV Programmer software interface. The interface is annotated with numbers 1 through 6, corresponding to the steps in the list above:

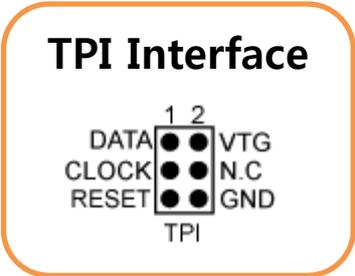
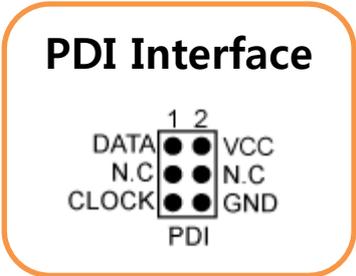
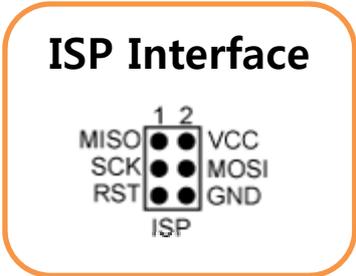
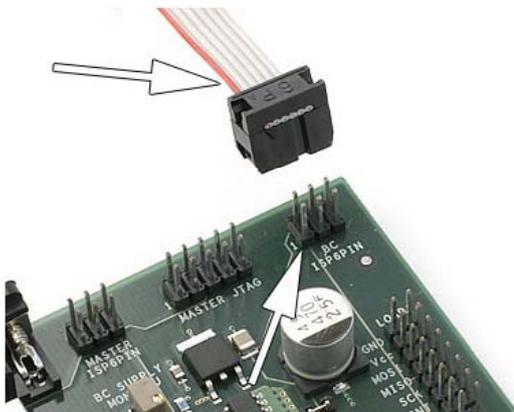
- 1**: Points to the **Comm Port** dropdown menu.
- 2**: Points to the **Device** dropdown menu, which is set to **ATmega128**.
- 3**: Points to the **Fuse 0xFD99E1** settings section, which includes options like **ATmega103 Compatibility Mode**, **Watchdog Timer**, **On-Chip Debug**, **JTAG Interface**, **Serial Program Downloading**, **CKOPT fuse**, **Preserve EEPROM memory**, **Boot Flash section size**, and **Boot Flash section size**.
- 4**: Points to the **Lock 0xFFFF** settings section, which includes options like **Mode 1: No memory lock**, **Mode 2: Further programming**, **Mode 3: Further programming**, and **Application Protection Mode**.
- 5**: Points to the **Open** button in the **Flash Data** section.
- 6**: Points to the **Transfer** button.

An **Open** dialog box is open in the center, showing a file named **sample.hex** in the **codes** directory. The **Files of type** is set to **a90 file(*.a90), hex file(*.hex)**. The **File name** is **sample.hex**. The **Files of type** dropdown is set to **a90 file(*.a90), hex file(*.hex)**. The **Open** button is highlighted with a red box.

Other visible elements include the **AVR GANG IV PROGRAMMER** logo, the **Ubiquitous generation PROCHILD** logo, the **ATMEL** logo, and the **Upgrade ProISPIV V1.00A** button.

Download & Target Writing

2. Connect ProISP IV to target board.



(Programming voltage for **TPI** is **5V** only)

3. Push the Programming Button.

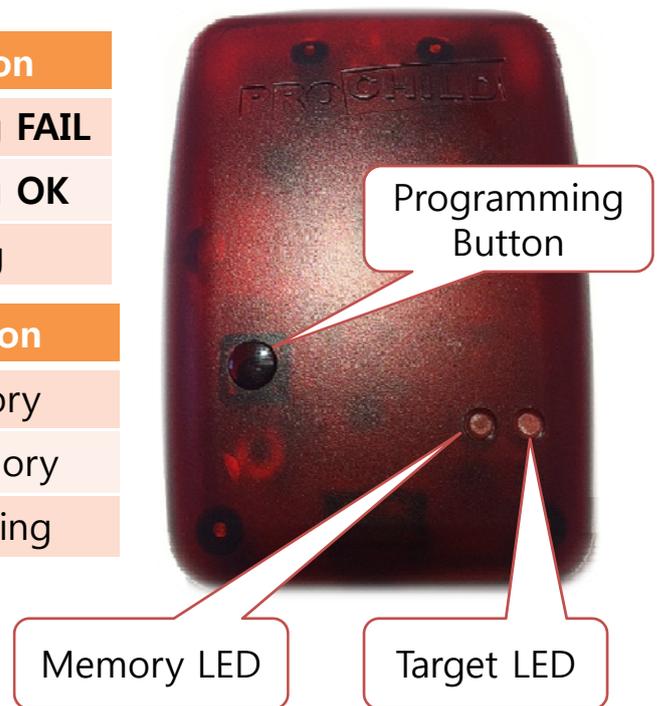
(The power for **ProISP IV** must come from **Target board**)

Hardware Description

1. Status LEDs

Target LED	Description
Red	Programming FAIL
Green	Programming OK
Orange blinking	Target writing

Memory LED	Description
Red	Empty memory
Green	Loaded memory
Orange blinking	Memory writing



2. Rear panel



Int. RC calibration bytes

- By changing the **OSCCAL** register from S/W, it is possible to get a higher calibration accuracy than by using the factory calibration.
- **ProISP IV** is designed to assign the user calibration value to the last two bytes of Flash memory. Therefore, the location of the data is determined by a Flash memory size. ([See below](#))

<Example>

- Device : ATmega16, 16KB flash memory
- Address : 0x3FFE, 0x3FFF
- Example source code

```
//Write user calibration bytes to OSCCAL  
asm("LDI    R30,    0xff"); //address of user data(0x3fff)  
asm("LDI    R31,    0x3F");  
asm("LPM                    ");  
asm("OUT    0x31,    R0 "); //address of OSCCAL(0x31)
```

Serializing

- **ProISP IV** can write a user **serial number** on the **internal EEPROM** of each target devices.

<How to use the **Serializing**>

<input checked="" type="checkbox"/> Serializing
EEPROM Start Address : 0x01
EEPROM End Address : 0x04
Initial Value : <input type="text" value="0x00000001"/>

- Check **Serializing**.
- The address of the serial number is **0x0001~0x0004**.
(**The address can not be changed.**)
- **Initial Value** is the first serial number of the target device.
- Whenever you write a target device, the **serial number** is increased by 1 automatically.