



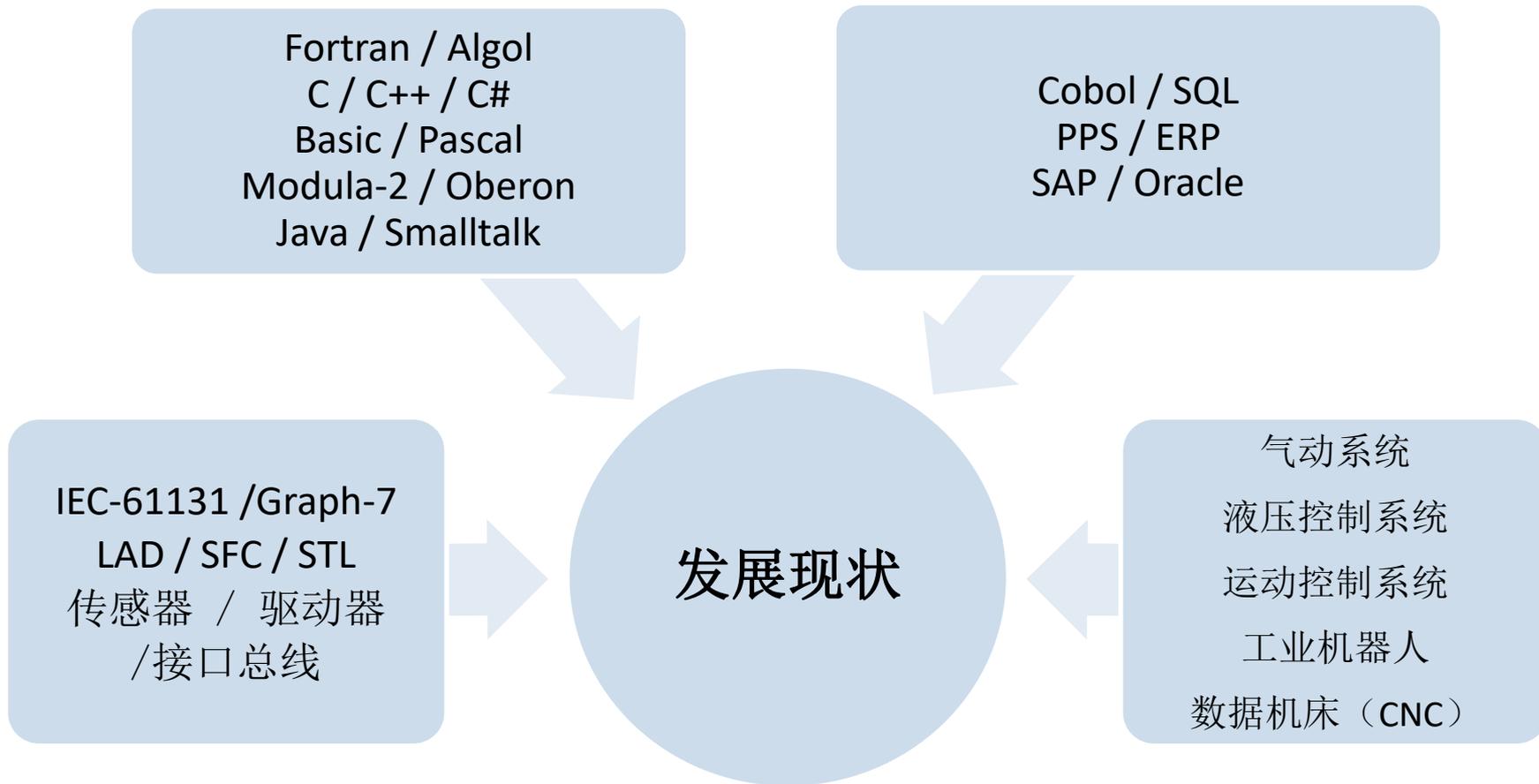
# The JetSym Tool

坚德自动化技术（上海）有限公司

修订日期：2018年6月15日

Revision date: June 15, 2018

# 发展历程



# Jetter编程：阶段1

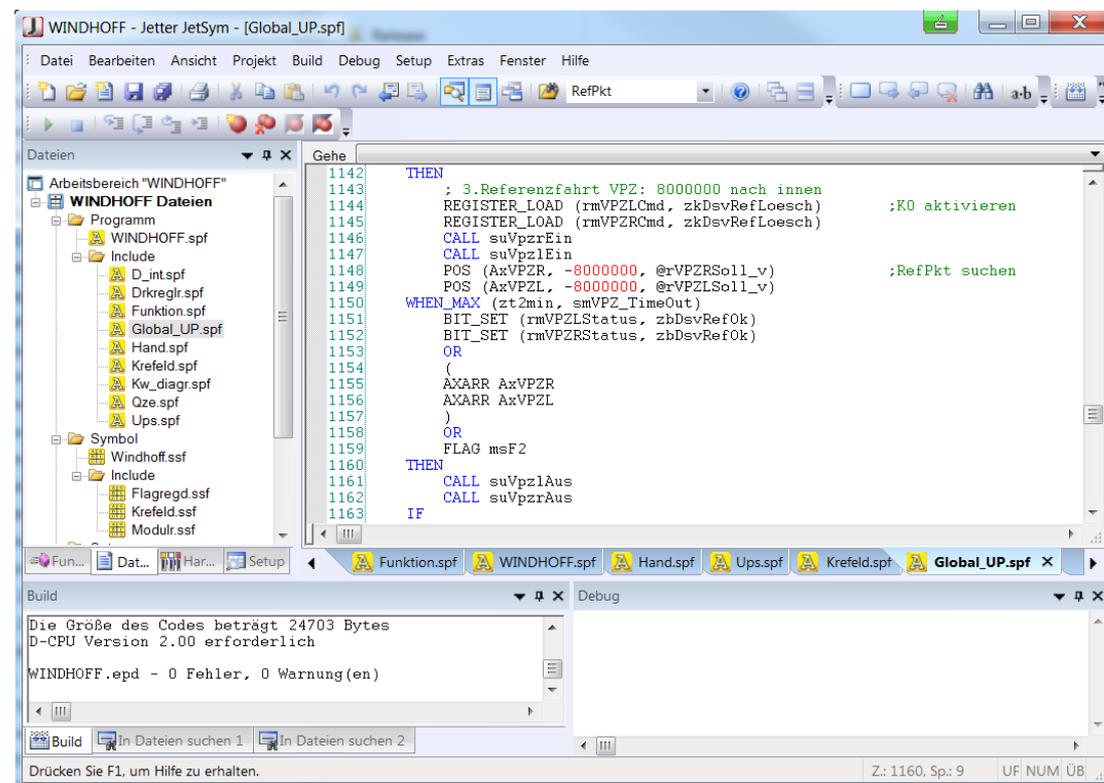
- 在DOS时代，Jetter控制器通过SYMPAS编程，SYMPAS即是编程工具的名称，同时也是编程语言的名称。

- ✓ 唯一
- ✓ 简易
- ✓ 纯文本格式
- ✓ 支持多任务
- ✓ 支持多语言
- ✓ 过程控制

```
Project File Edit Block Transfer Listing Monitor Scope Special
===== Length: 1382 (17.1%) ===== UPS.PPD =====
573:      THEN
574:          ; 3.Referenzfahrt UPZ: 8000000 nach innen
575:      REGISTER_LOAD [rmUPZLCmd with zkDsvRefLoeschl ;K0 aktivieren
576:      REGISTER_LOAD [rmVPZRCmd with zkDsvRefLoeschl
577:      CALL suUpzrEin
578:      CALL suUpzIEin
579:      POS [axis=AxUPZR, pos=-8000000, v=R(rUPZRSoll_v)] ;RefPkt suchen
580:      POS [axis=AxUPZL, pos=-8000000, v=R(rUPZLSoll_v)]
581:      WHEN_MAX [max.time=zt2min, subroutine=smUPZ_TimeOut]
582:      BIT_SET [reg=rmUPZLStatus, bit=zbDsvRefOk]
583:      BIT_SET [reg=rmUPZRStatus, bit=zbDsvRefOk]
584:      OR
585:      <
586:      AXARR axis=AxUPZR
587:      AXARR axis=AxUPZL
588:      >
589:      OR
590:      FLAG msF2
591:      THEN
592:      CALL suUpzIAus
593:      CALL suUpzrAus
594:      IF
F1 Help Shift-F9 Syntax-Check Ctrl-F9 Transmit F10 Menu DELTA
```

# Jetter编程：阶段2

- 基于windows的后续版本称为JetSym，它既代表一种编程语言，同时也代表这一代控制器的编程环境。



# Jetter编程：阶段3

- 参考ST编程语言标准IEC-61131-3，开发出了编程语言JetSym ST。

```
1 //
2 // automatic mode
3 // Order of movements:
4 //
5 task t_Auto
6
7 Label lb_Auto:
8
9
10     when bf_Auto continue;
11
12     sr_ResetManual;
13
14     if not bf_Reference then //no automatic without reference
15         bit_set(nm_Err_2,cb_ErrRef);
16         delay(ct_1s);
17         bf_Auto:=false;
18         goto lb_Auto;
19     end_if;
20
21 //-----
22 // prepare automatic run
23
24     bv_DataChanged:=true; //calculate the values
25     when not bv_DataChanged or not bf_Auto continue;
26
27     //move finger to starting point
28     //
29     MotionMovePtp ( Finger, All, Absolute, fm_Fng_StartPoint, 600, 20000, 20000, * )
30
31     //to ensure te correct position
32     MotionMovePtp ( Spreader, All, Absolute, 0.0, 700, 20000, 20000, * )
33
34     when
35         MotionReadStatus ( Spreader, Angehalten )
36         and
37         MotionReadStatus ( Finger, Angehalten )
38     continue;
39
40
```

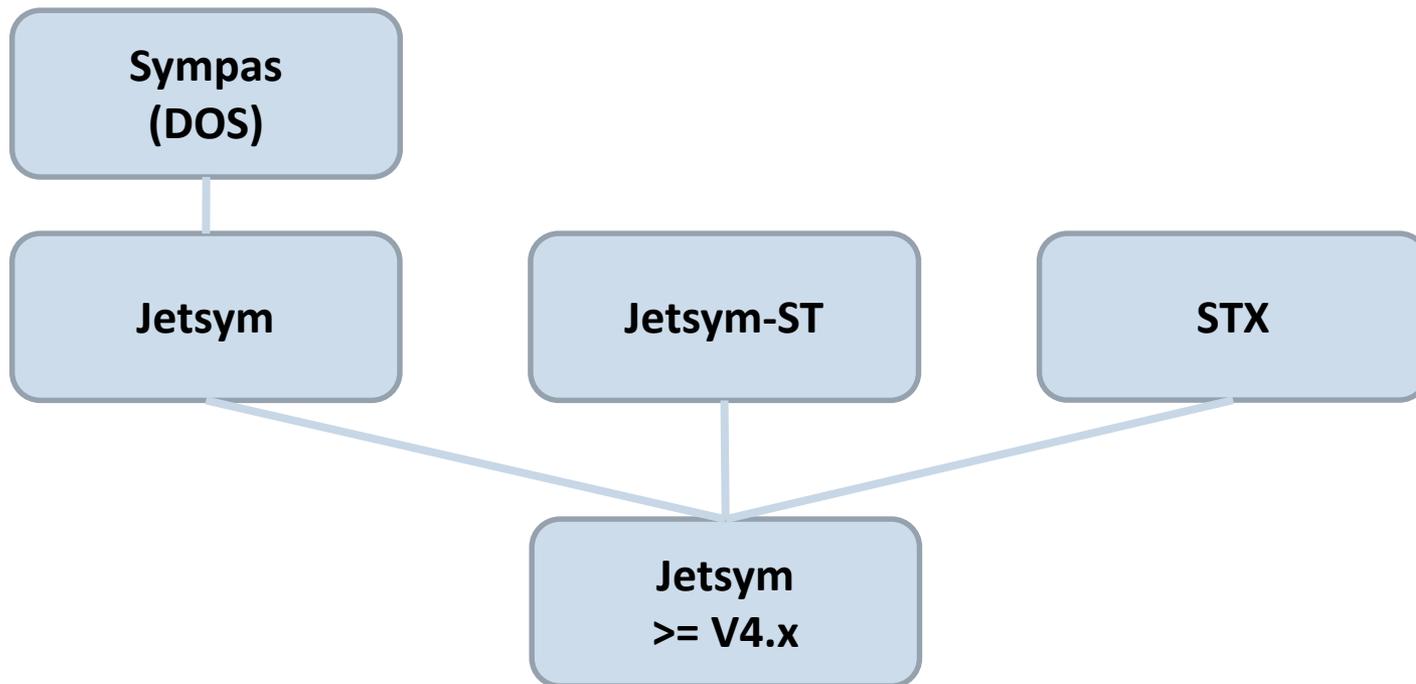
# Jetter编程：阶段4

在开发新一代控制器的过程中，ST语言也经过了彻底的重新设计和增强。STX语言应运而生。

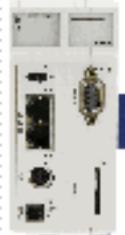
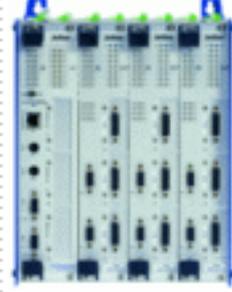
```
1 // Load control Task
2
3 TASK T_Load;
4
5     Loop:
6
7         When bf_RunActive and bf_ReStart continue;
8         TimerStart(nt_Load, t#10s);
9
10        // Spring Transfer
11        TimerStart(nt_XFer, T_Timeout);
12        bo_SpringEngsOut := true;
13
14        // Wait for spring fingers to be out
15        When not bi_SprFngHome continue; //ijw 2/9/09
16        If TimerEnd(nt_XFer) then
17            Bitset(n_MCErr, 10);
18        End_if;
19        bo_SpringCompDown := true;
20
21
22        // Check finger isn't trapped
23        When bi_SprFngOut or TimerEnd(nt_XFer) continue;
24        TimerStart(nt_XFer, T_Timeout);
25
26        // Wait for spring compressor to be down
27        When bi_SprCompDown or TimerEnd(nt_XFer) continue;
28        If TimerEnd(nt_XFer) then
29            Bitset(n_MCErr, 11);
30        End_if;
31        bf_ReStart := false; // Reset Restart Flag Now
32
33        // Wait for cloth move to be complete
34        When bf_ClothDone continue;
35        bf_ClothDone := false;
36
```

# 功能整合

JetSym从版本V4.00开始，所有用于Jetter控制系统的编程语言都被合并到一个软件包中。



# 控制系统和编程语言发展过程

JetSym	JetSym ST	STX
<p data-bbox="202 506 347 542">PASE-E</p> 	<p data-bbox="705 621 946 656">JetControl 24x</p> 	<p data-bbox="1275 606 1516 642">JetControl 3xx</p> 
<p data-bbox="202 806 299 842">Nano</p> 	<p data-bbox="705 899 946 935">JetControl 647</p> 	<p data-bbox="1275 921 1516 956">JetControl 9xx</p> 
<p data-bbox="202 1085 299 1120">Delta</p> 		

# JetSym编程软件

---

- 安装并启动JetSym
- 获取授权许可
- 工作空间
- GUI界面
- 客户定制
- 创建并配置项目
- 硬件管理
- 上传/下载 | 在线比较

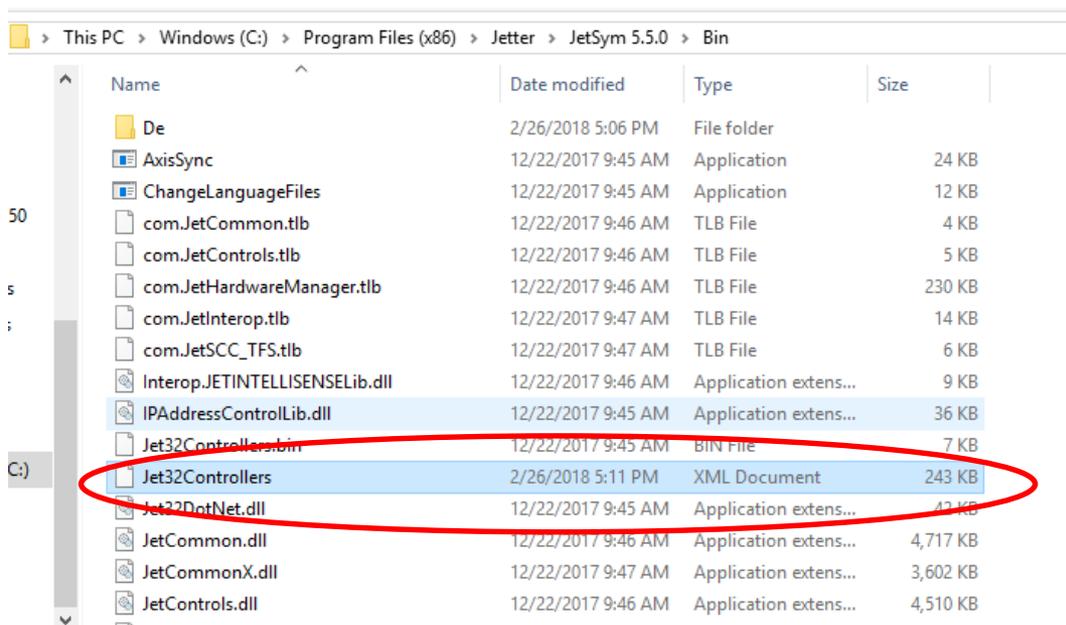
# JetSym 安装

---

- 软件下载: <https://www.jetter.de/en/downloads/software/jetsym.html>;
- 选择安装语言: 英语/德语;
- 安装向导指导用户完成安装过程。
- JetSym 用户界面语言, 根据个人电脑的操作系统选择英语/德语。
- JetSym v5.0 及后续版本中, 可以同时安装多个版本。

# 激活JC-120MC控制器

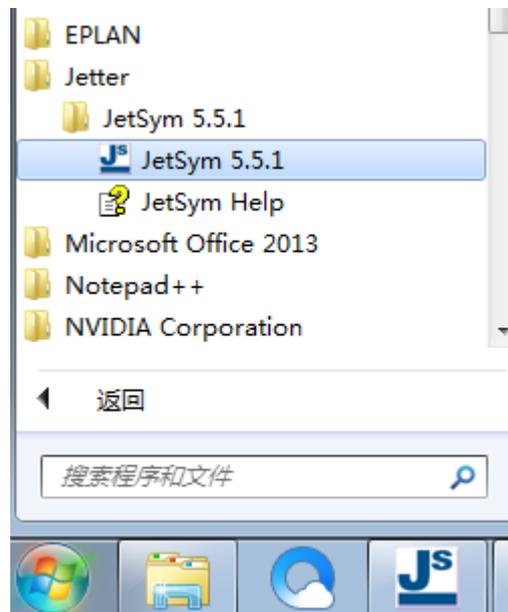
- 打开JetSym安装目录下的Bin文件夹；
- 打开文件：Jet32Controllers.XML；
- 搜索JC-120MC控制器；
- 将“<Disabled>”修改为“0”；
- 保存文件；
- 关闭并重新打开JetSym；



```
6489 </Controller>
6490 <Controller>
6491   <Name>JC-120MC</Name>
6492   <Index>55</Index>
6493   <Disabled>0</Disabled>
6494   <General>
        <FirstVersion>100</FirstVersion>
```

# 启动JetSym

- 开始菜单
- 桌面
- 任务栏
- 快捷方式



## 注意:

- 1、您还可以通过双击工作空间或项目文件来很容易地启动JetSym。
- 2、如果已经安装了几个版本的JetSym，那么只有一个版本可以注册到JetSym文件中。您可以很容易地更改文件注册：Menu: Tools/Register JetSym version

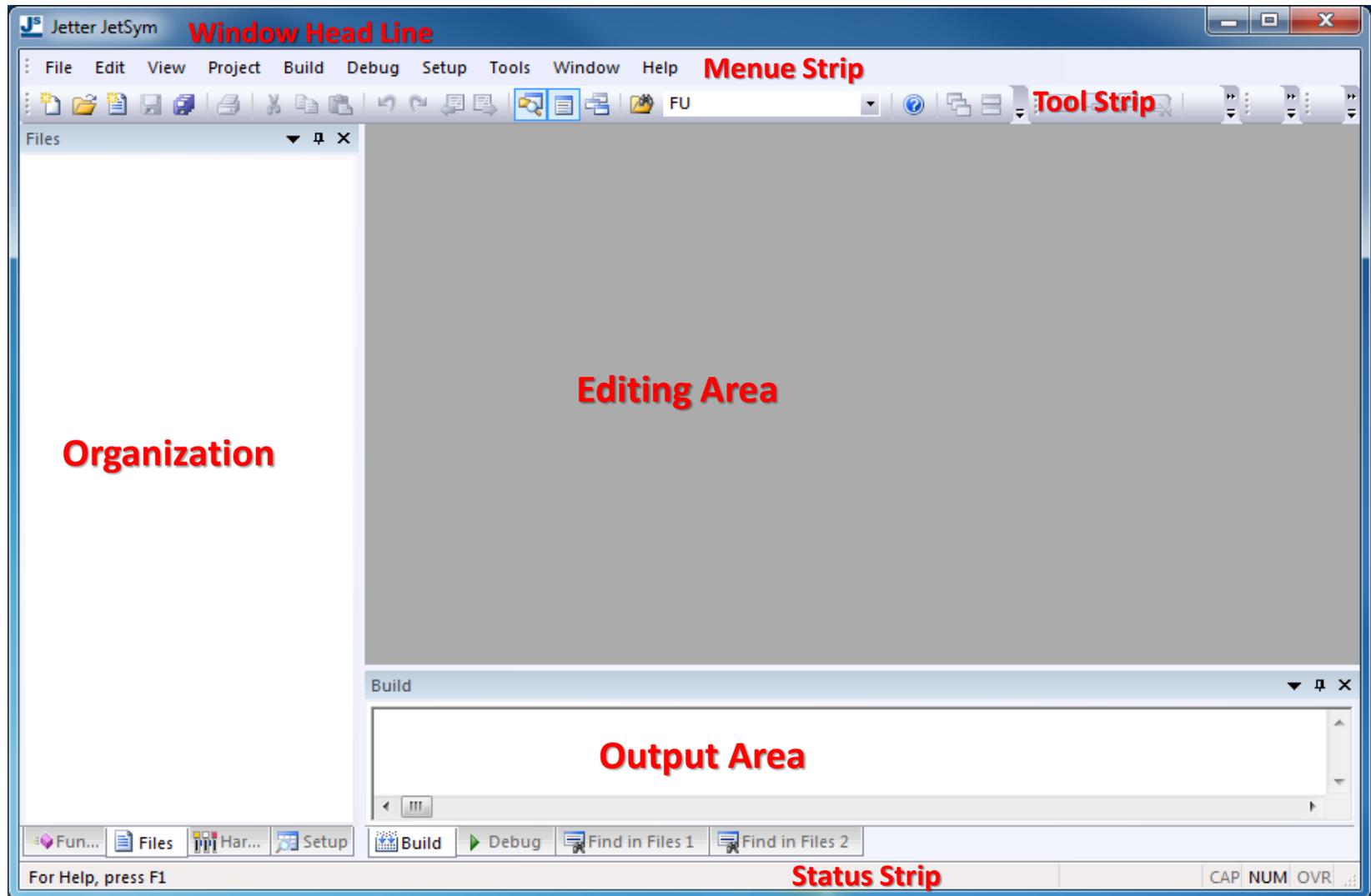
# 授权许可

- 如果没有授权许可，最多可创建4KB的编译代码；
- 安装授权时，须以管理员方式打开JetSym；
- 安装授权：**Menu: Help/Register JetSym**；
- 请参阅相关授权文件；

## Build

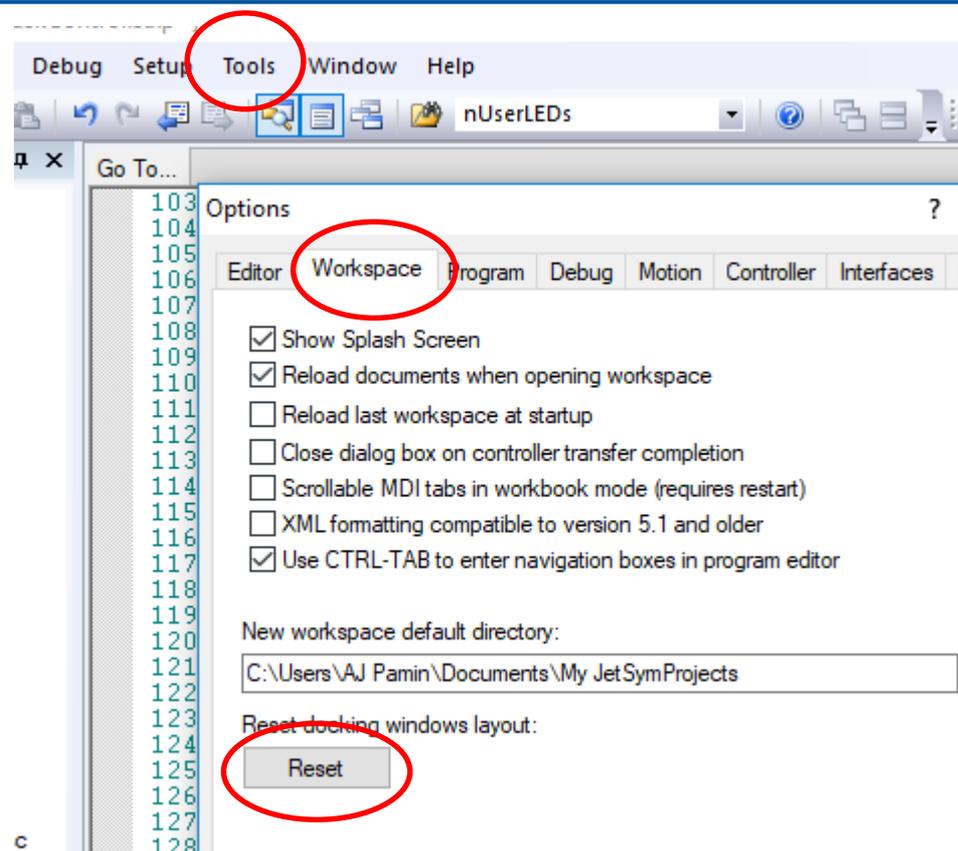
```
>> Event.stxp
>> Zylinder.stxp
>> Netzwerk.stxp
>> Visu.stxp
Optimizing code...
Assembling code...
>> Code size: 4963 bytes.
>> Data size: 92 bytes.
>> Stack size: 24576 bytes.
D:\Schulung\STX_Basissch...\JV_STX.stxp(7) : error 3031: Program size exceeds limit (4096 bytes) - check license.
JV_STX - 1 error(s), 0 warning(s)
```

# JetSym用户界面



# 重置默认工作区域窗口

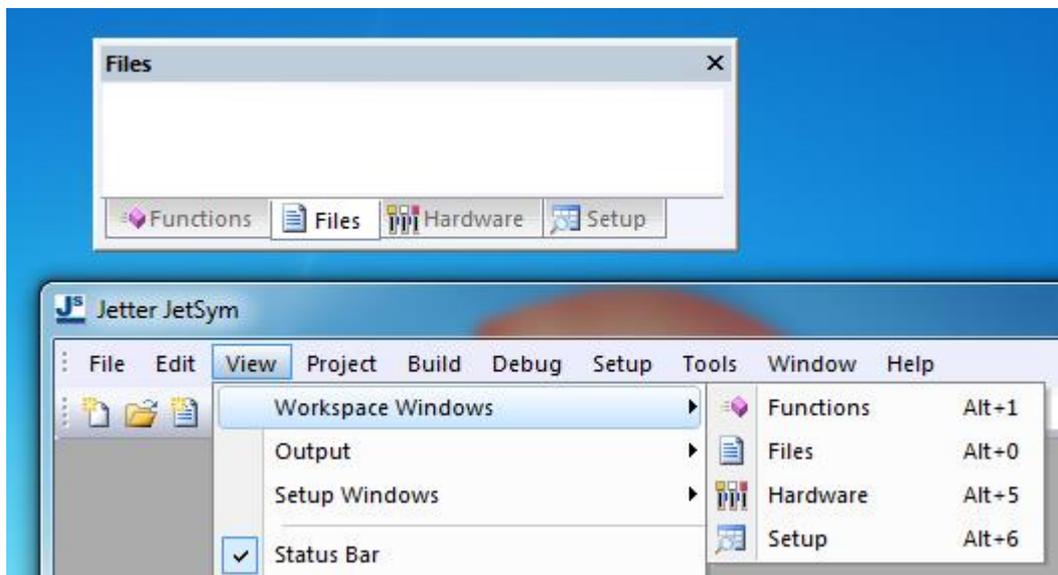
- 1、如右图，点击“Reset”  
(Tools>Options>Workspace)；
- 2、关闭并重新打开JetSym；



# 组织区

- 基本设置提供了以下类别的选项卡：

- Files
- Functions
- Hardware
- Setup

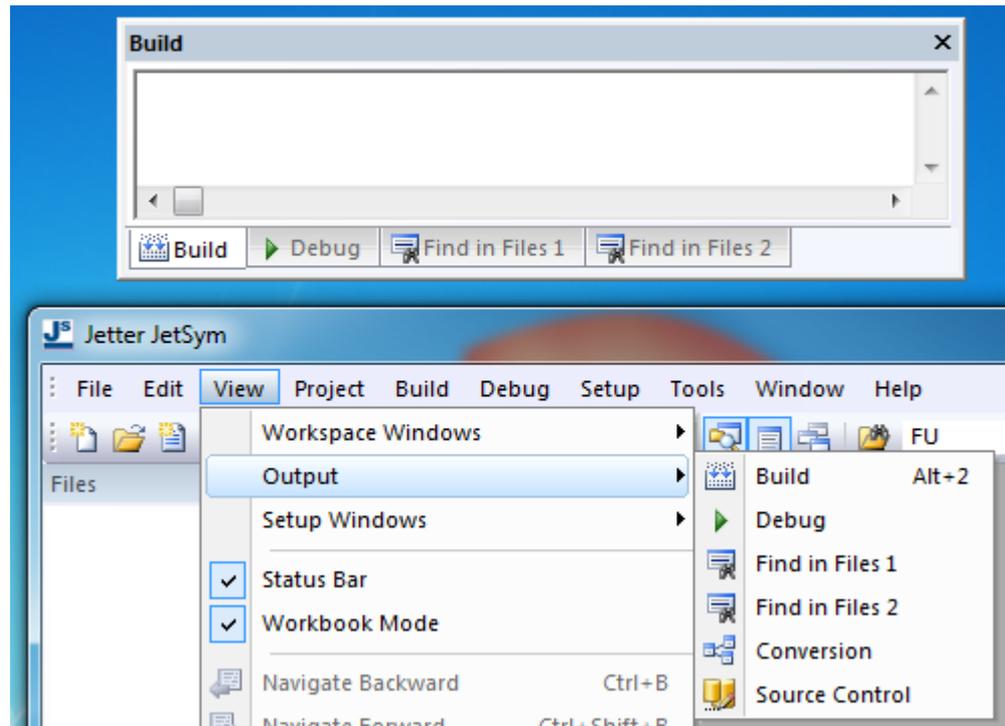


- 用户也可以自定义组织区：

- 单独打开/关闭；
- 动态显示/隐藏；
- 锁定/浮动；

# 输出区

- 用户自定义；
- 输出窗口中显示的基本设置引用以下类别：
  - Build
  - Debug
  - Find in Files 1
  - Find in Files 2

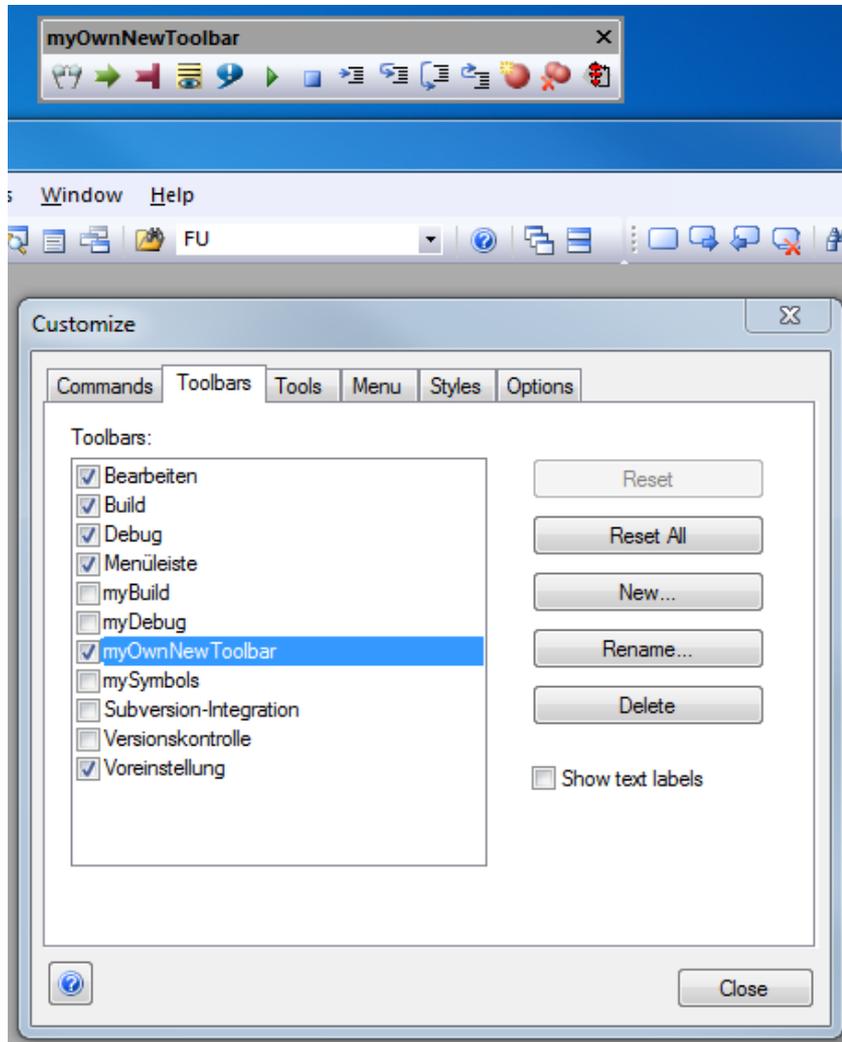


# 编辑区

---

- 在这个窗口中，会显示所有活动窗格（文件）。数据类型决定显示方式。
- 如果您激活了工作簿模式视图，那么所有打开的窗口都将由对应的选项卡表示。可以通过拖动选项卡显示多个窗口。

# 工具栏



- 固定/浮动
- 显示/隐藏
- 重置
- 重命名
- 创建/自定义工作栏

# JetSym常用快捷键

---

- Ctrl+N（新建）、Ctrl+O（打开）、Ctrl+S（保存）、Ctrl+P（打印源码）
- Ctrl+F（查找）、Ctrl+Shift+F（文件中查找）
- Ctrl+H（替换）、Ctrl+Shift+H（文件中替换）
- Ctrl+Alt+T（自动插入）
- Ctrl+Alt+P（显示参数）
- Ctrl+J（自动补全关键字）
- Alt+鼠标拖动（矩形框选）
- Alt+Shift+C（批量注释）、Alt+Shift+U（取消批量注释）
- FN+F1/FN+Shift+F1（打开JetSym帮助文档）

# 什么是JetSym项目？

---

- 对于一个单独的控制器，一个项目由以下部分组成：
  - 配置文件；
  - 程序和库；
  - 数据文件；
  - 文档；
- 将所有必需的文件保存在一个目录中，有利于项目的归档、恢复和共享等；
- 一个项目包含多个配置文件；



# 什么是JetSym配置文件？

---

- JetSym的配置文件表示针对硬件的设置；
- 配置文件用来管理JetSym项目的变量和选项目；
- 在条件编译的帮助下，您可以在程序中管理不同的配置文件。
- 如果需要修改配置，应断开JetSym与控制器的连接，修改完成后再次下载。
- 激活的配置为粗体显示。



# 什么是JetSym的工作空间?

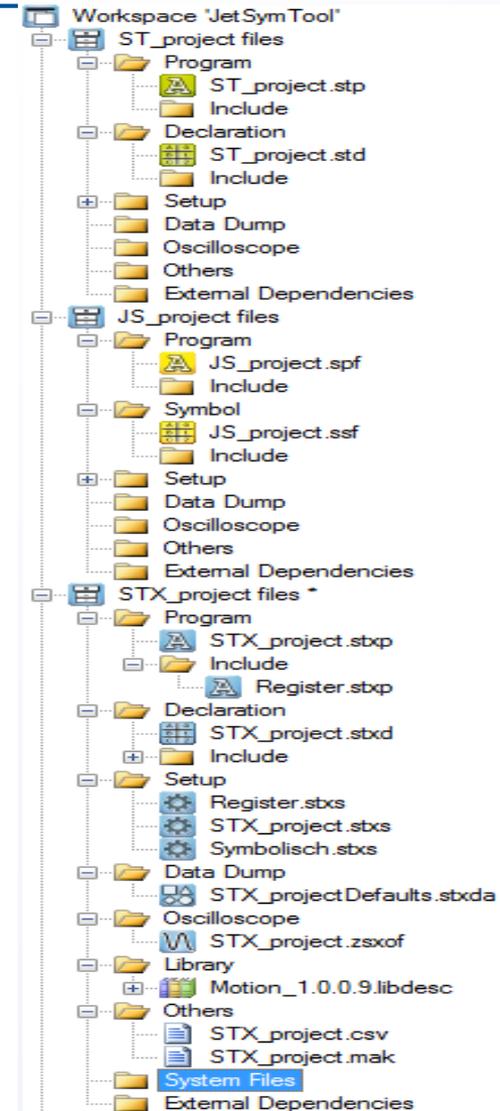
---

- 工作空间是一个对多个JetSym项目进行组织和管理的结构;
- 在工作空间中,同时只能有一个项目被激活;
- 激活的项目以粗体显示;
- 全局命令,如编译或传输命令,与激活项目相关;
- 一个项目也可以属于几个工作空间;
- 为了改变激活项目,必须与控制器(监视器、安装程序或调试器)断开通讯链接;



# 项目文件树 (STX)

	Ending	Description
	*.stxp	Source text/program files
	*.stxd	Declarations table/symbols
	*.stxs	Setup/online values
	*.stxda	Data files/data backup
	*.sxof	Oscilloscope

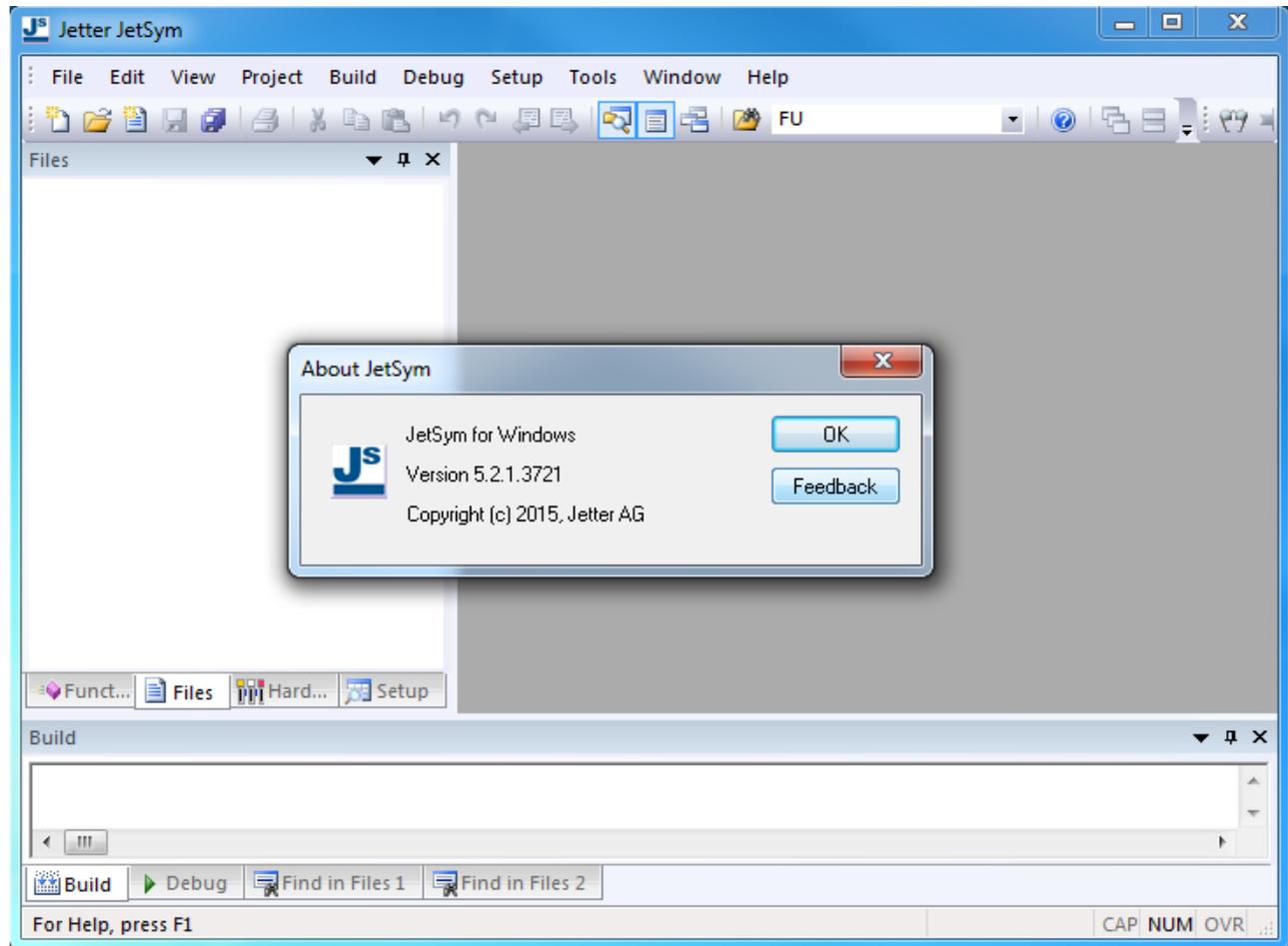


# 项目文件（JetSym | ST | STX）

File	STX	JetSym-ST	JetSym
Program	*.stxp 	*.stp 	*.spf 
Declaration	*.stxd 	*.std 	*.ssf 
Setup	*.stxs 	*.sts 	*.ssu 

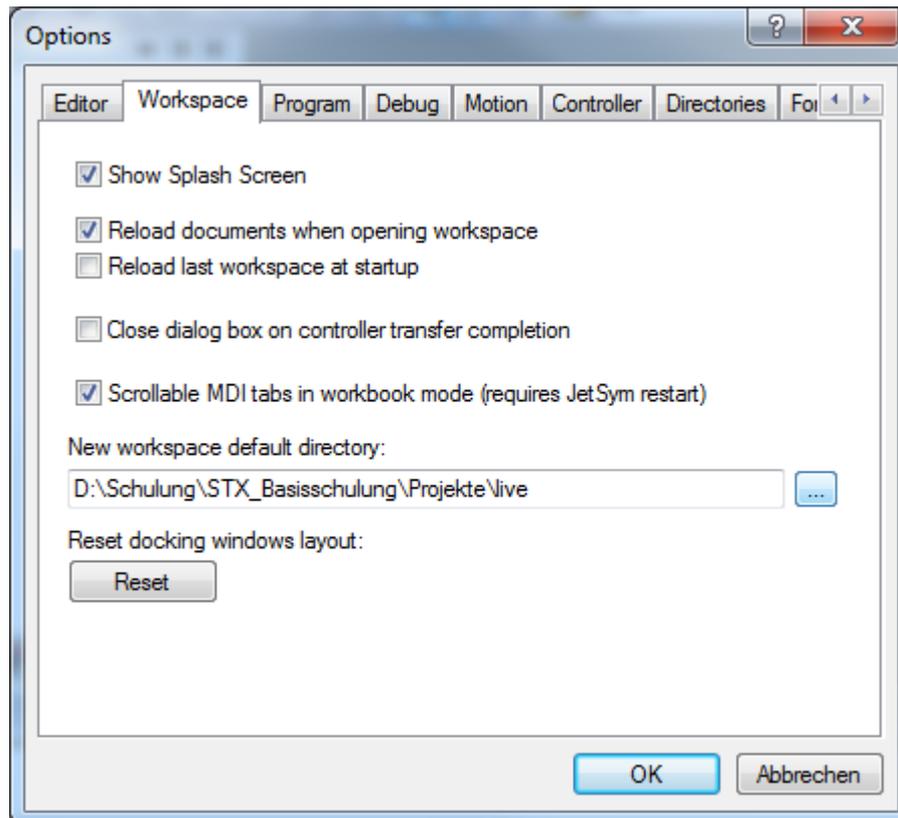
# 检查JetSym版本

- Menu: Help → About JetSym...



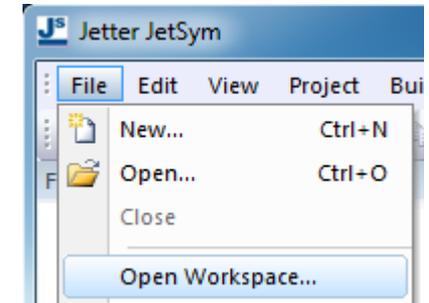
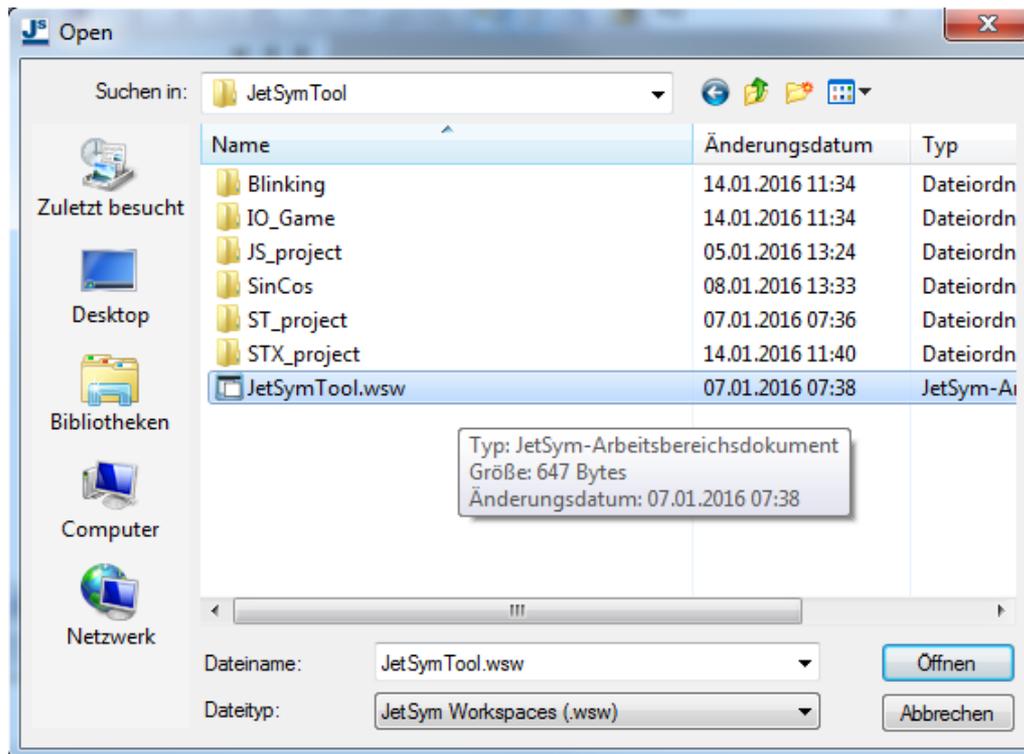
# JetSym 设置

- Menu: Tools → Options...

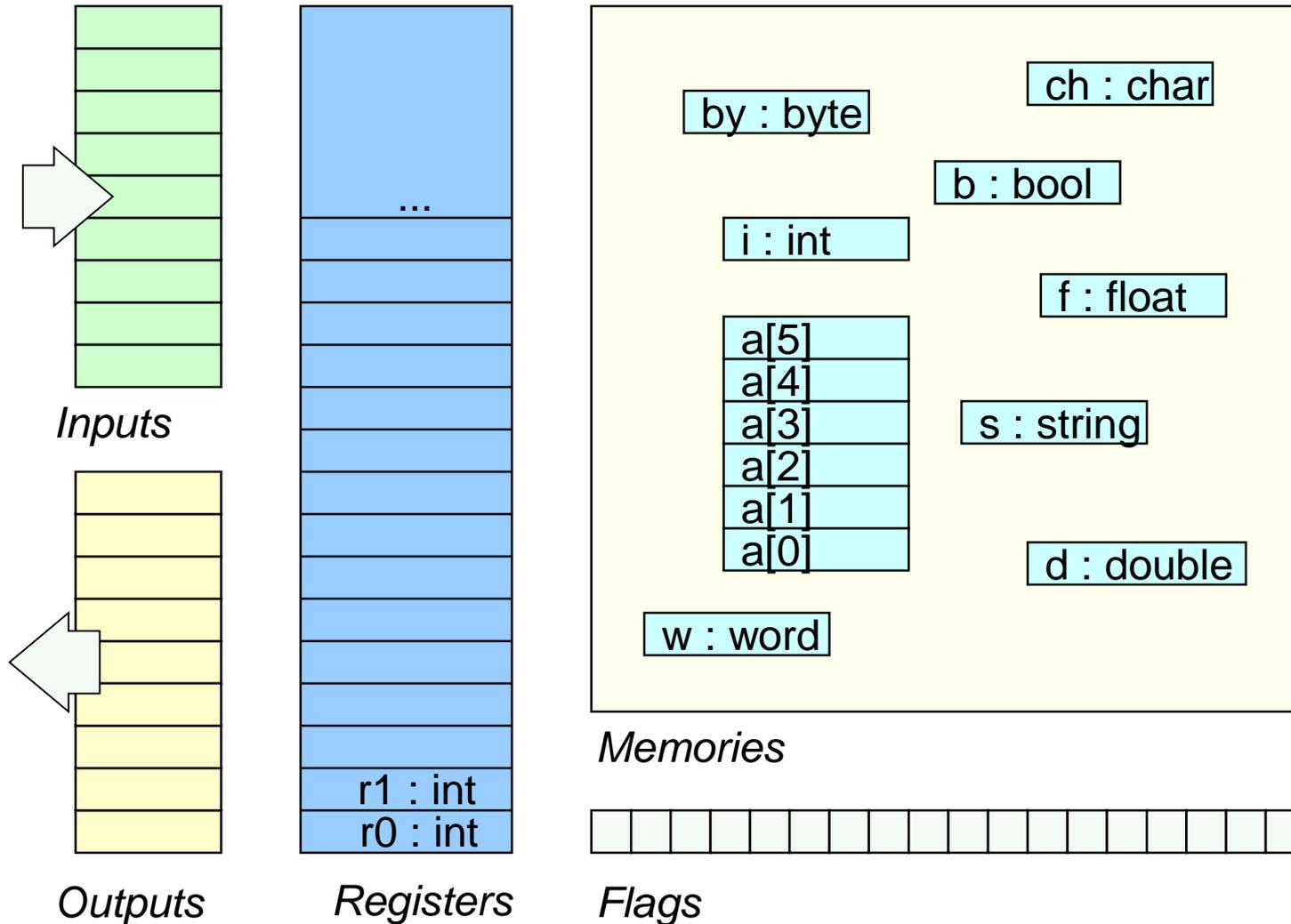


# 打开工作空间

- 工作空间的文件后缀为 “.wsw” ；



# 寄存器变量和内存变量



# Setup-显示数据

The screenshot displays the JetSymbTool IDE interface. On the left, a project tree shows the 'Setup' folder containing files like 'Register.stxs', 'STX\_project.stxs', and 'Symbols.stxs'. The main editor window shows the source code for the 'Setup' task, including tasks for 't\_0', 't\_Blink', 't\_Lamp\_AB', and 't\_Lamp'. The code includes comments and task definitions with loops and delays.

Name	Number	Content
t_0	0: line 4	task broken
t_Blink	1: line 11	delay
t_Lamp_AB	2: line 21	delay
t_Lamp	3: line 36	delay
t_Flash	4: line 59	running
t_Register	5	task broken
t_Flags	6: line 84	delay
//Eingänge		
iSwitch1	100000201	false
iSwitch2	100000202	false
iSwitch3	100000203	false
iSwitch4	100000204	false
iButton1	100000205	false
iButton2	100000206	false
iButton3	100000207	false
iButton4	100000208	false
//Ausgänge		
oBlink	100000209	false
oLamp_A	100000210	true
oLamp_B	100000212	true
oLamp_1	100000212	true
oLamp_2	100000213	false
oLamp_3	100000214	false
oFlash	100000215	false
oNotUsed	100000216	false

# 寄存器类型变量

寄存器类型 Register type	存储位置 Location	描述 Comment
系统寄存器 System register	CPU	系统参数； 实时时钟 (RTC), EDS, 特殊的功能及接口；
模块寄存器 Module register	Modules	参数； 实际值（开关量/模拟量）； 状态信息； 指令；
用户寄存器 Application register	SRAM	非易失性； 用于通讯；

# 寄存器实例

- 所有控制器手册中都有对应的快速参考（Quick reference）
- 下面仅以JC-120MC为例

模块	寄存器	描述
RTC	102911, ff.	Seconds
SD	107000, ff.	SD memory card
Switch	108015	Mode selector
JX2 bus	2000mmxxx	Servo axis
JX3 bus	100mmxxxx	DIO16, TH12
Error	200008	Error register 1
SRAM	1000000, etc.	Int, Float, RegString

## JC-120MC

### 15 快速参考 - JC-120(MC)

#### 相应的操作系统版本

本快速参考简要总结了操作系统版本为 1.26.1.00 的 JC-120 和 JC-120MC 控制器的寄存器和标记。此外，还描述了插入式连接器的分配，及通过 DIP 开关设置 IP 地址。

#### CANopen® 总线上的默认地址

JC-120(MC) 的出厂默认地址为：  
节点 ID: 127 (0x7F)

#### CANopen® 接口的最大数量

JC-120(MC):  
CAN 接口 (CANopen®-STX-API) 的最大数量: 2  
CANMAX: 1

#### 以下操作系统版本支持心跳监控 (CANopen®-STX-API)

JC-120(MC):  
操作系统版本: 1.26.1.00

#### 一般概述 - 寄存器

100000 ...100999	电子数据表 (EDS)
101000 ... 101999	配置
102000 ... 102999	实时时钟 (RTC)
103000 ... 103999	串行接口
104000 ...104999	以太网
107000 ...107499	SD 存储卡
107500 ...107599	闪存盘
108000 ...108999	CPU/基板
200000 ...209999	一般系统寄存器
210000 ...219999	应用程序
220000 ...229999	HMI 控制
230000 ...239999	通过 JetIP 联网
240000 ...249999	JetSync
250000 ...259999	以太网系统总线

# Setup-变量类型

Type	Description	Data
int	Register display as integer	整型
float	Register display as floating point number	浮点型
auto	int/float	整型/浮点型
bool	Logical value	布尔量
input	Digital inputs	开关量输入
output	Digital outputs	开关量输出
regstring	Text display	字符串
task	State and line	当前任务行

# 通过寄存器访问硬件

---

- 寄存器宽度：32位（4字节）；
- 寄存器类型：整型/浮点型；
- 寄存器服务于硬件，变量服务于程序；
- 在运行时（连接中断），读写寄存器也会失败！
- 寄存器不能作为内存访问功能，它们是硬件接口概念的一部分！

# 通过变量表创建符号变量

- 用户寄存器
- 系统寄存器
- 模块寄存器
- 输出
- 输入
- 标志位
- 常量
- 文本
- 定义默认值

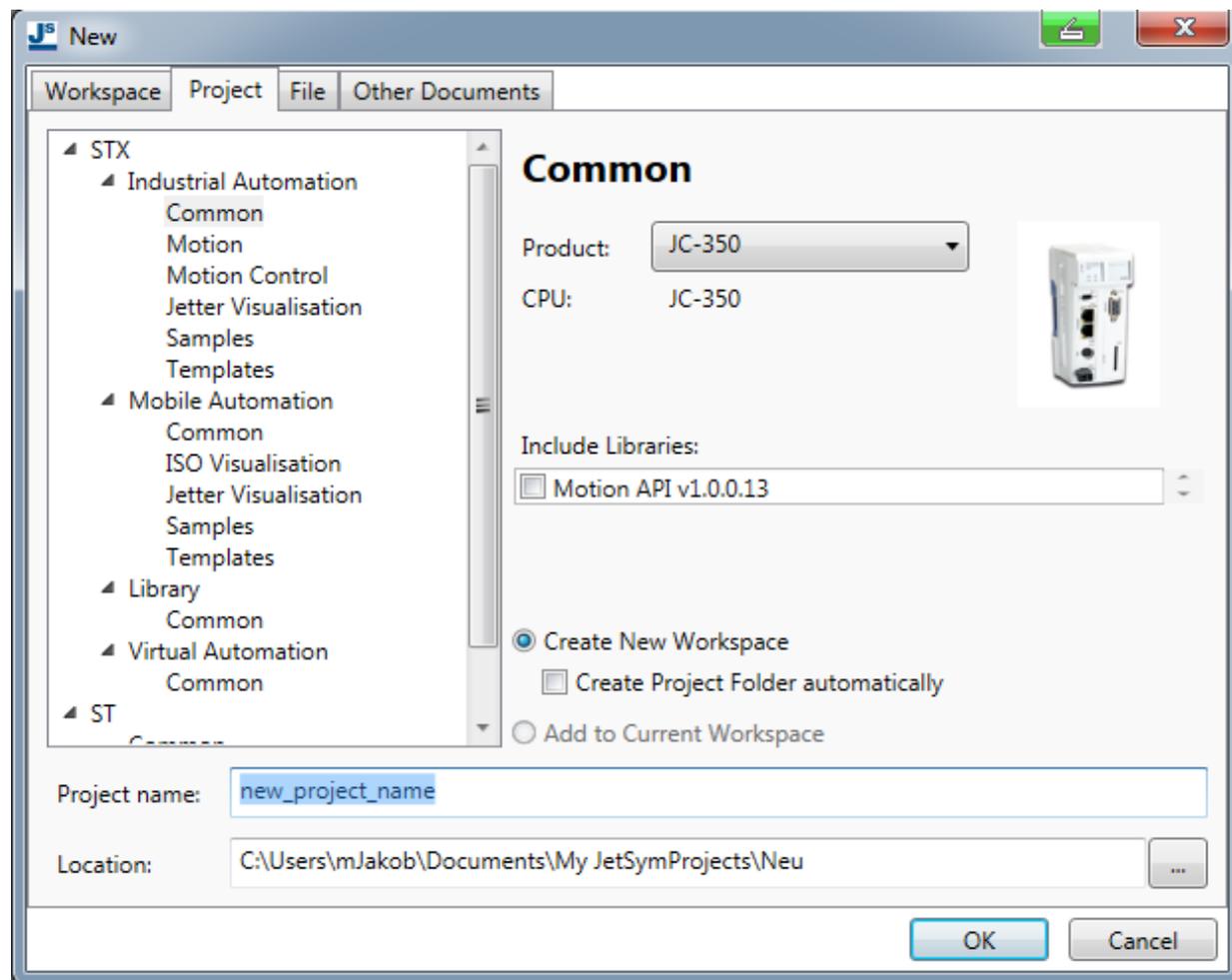
	Name	Number	Type	Default	Comment
13	nSum	1000000	int	0	
14	nStep	1000001	int	1	
15	rsSec	102911	int		
16					
17	b_1	1	bool		flags[x]
18	b_2	2	bool	false	
19	b_3	3	bool	false	
20	b_4	4	bool	true	
21					
22	Value100	1000100	int		mem[x]
23	Value101	1000101	int	101	
24	Value102	1000102	int	102	
25	Value103	1000103	int	103	
26					
27	i201	100000201	input		inputs[x]
28	o209	100000209	output		outputs[x]

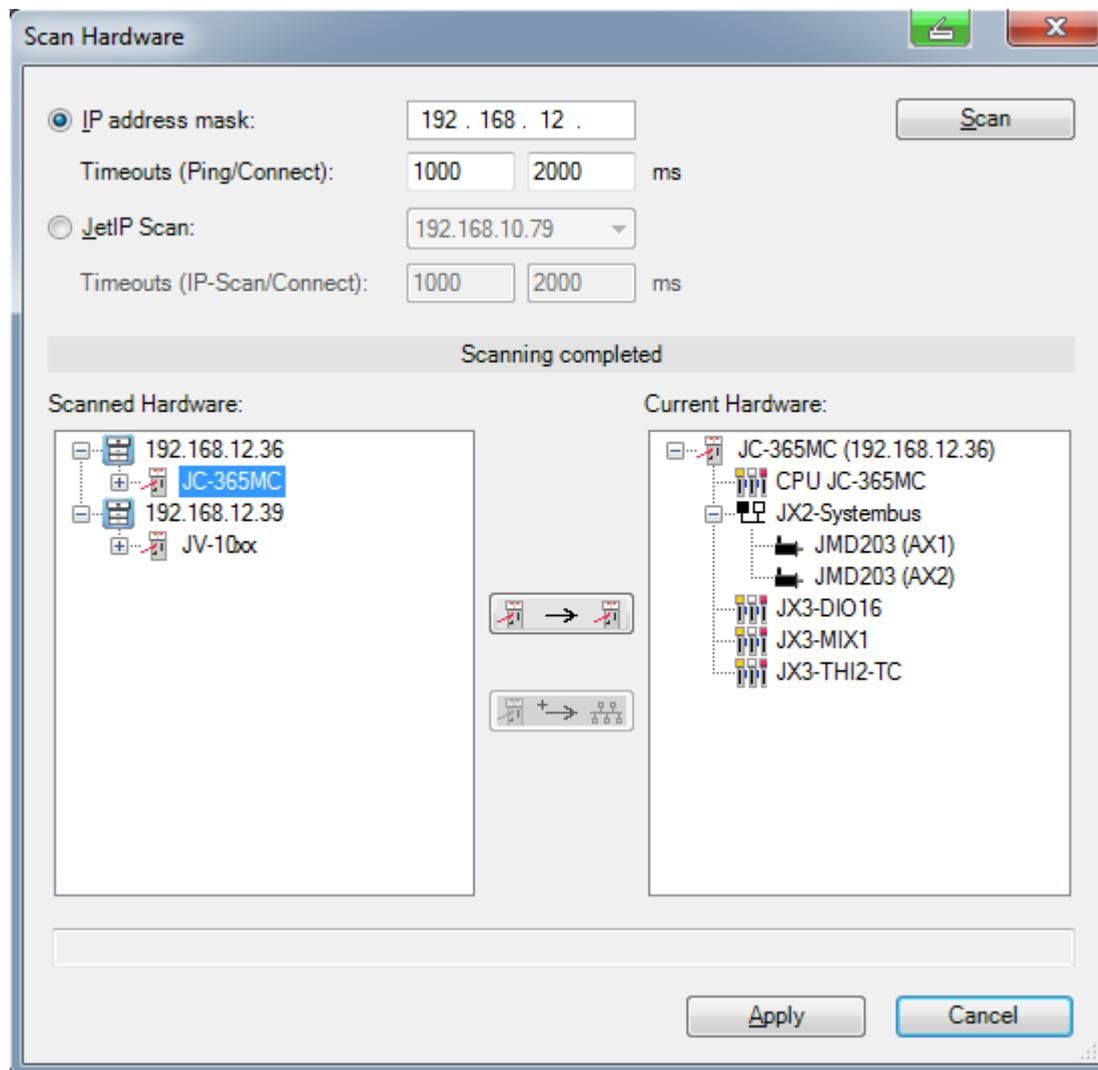
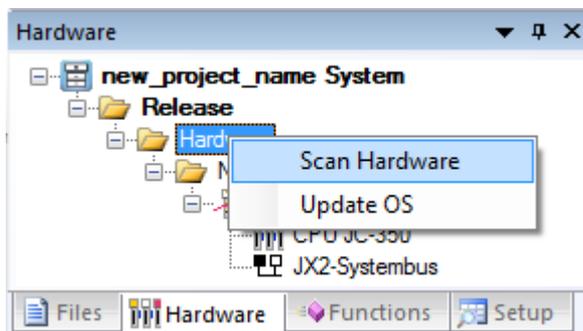
	Name	Address	Content	Type	Copie
1	nSum	1000000	0	regINT	
2	nStep	1000001	1	regINT	
3	b_2	2	0	FLAG	
4	b_3	3	0	FLAG	
5	b_4	4	1	FLAG	
6	Value101	1000101	101	regINT	
7	Value102	1000102	102	regINT	
8	Value103	1000103	103	reqINT	

# 创建新项目

- 项目类型
- 控制器选择
- 库
- 工作空间
- 项目名称
- 项目路径



# 硬件扫描



# 网络配置

## ■ 以太网IPv4设置(PC)

- 设置TCP/IPv4的IP地址和子网掩码，使控制器与个人电脑IP处于同一网段
- 通过测试(test)按钮在线测试通讯是否正常



Internet 协议版本 4 (TCP/IPv4) 属性

常规

如果网络支持此功能，则可以获取自动指派的 IP 设置。否则，您需要从网络系统管理员处获得适当的 IP 设置。

自动获得 IP 地址 (I)

使用下面的 IP 地址 (S):

IP 地址 (I): 192 . 168 . 1 . 99

子网掩码 (U): 255 . 255 . 255 . 0

默认网关 (G):

自动获得 DNS 服务器地址 (I)

使用下面的 DNS 服务器地址 (S):

首选 DNS 服务器 (P):

备用 DNS 服务器 (A):

退出时验证设置 (L)

高级 (A)...

确定 取消

Controller

Product: JC-120MC

CPU: JC-120MC

Version: 1.29 Test

Autorun  AutoFlash  Online Version Detection

Interface

Type: Ethernet

IP address: 192.168.1.3

Timeout: 5000 ms

Baudrate:

Port number: default

# 控制器IP地址设置

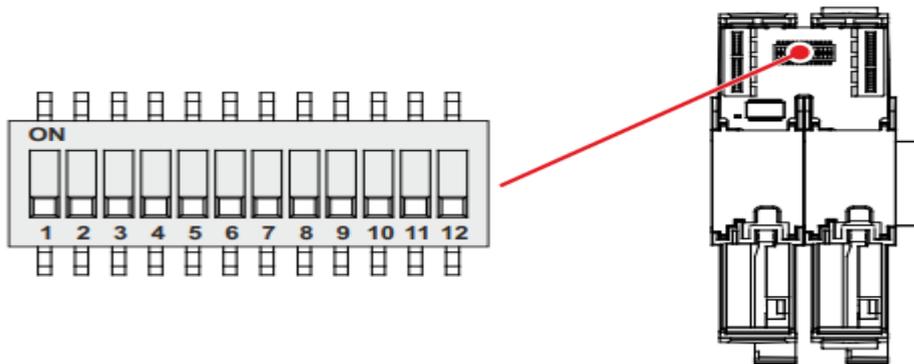
---

- IP可以通过如下方式重新设置
  - 重新恢复出厂设置（出厂默认IP：192.168.10.15）
  - 通过配置文件（ System/Config.ini ）
  - 通过配置文件（ System/Config.ini ）和拨码开关（DIP）
  - 通过非易失性寄存器
  - 运行期间修改IP地址

# IP地址恢复出厂设置

## 步骤:

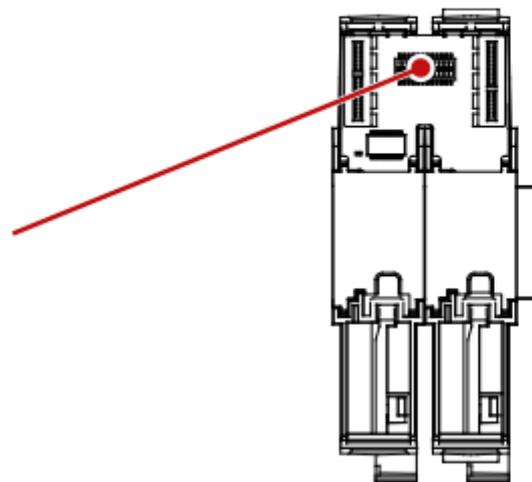
- 将拨码开关（DIP）按下图设置
- 重启控制器
- 出厂默认IP地址：192.168.10.15



# 通过配置文件设置IP

## 步骤:

- 通过FTP建立控制器与PC的连接;
- 以管理员权限登录, 并输入用户名和密码  
用户名: admin;  
密码: admin;
- 打开System文件夹
- 修改并保存config.ini文件
- 关毕FTP连接
- 重启控制器



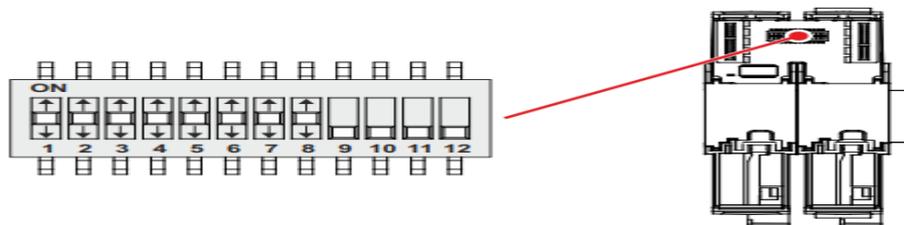
# 通过配置文件和拨码开关设置IP

## 步骤:

- 通过FTP建立控制器与PC的连接;
- 以管理员权限登录, 并输入用户名和密码  
用户名: admin;  
密码: admin;
- 打开System文件夹
- 修改并保存config.ini文件
- 关毕FTP连接
- 重启控制器

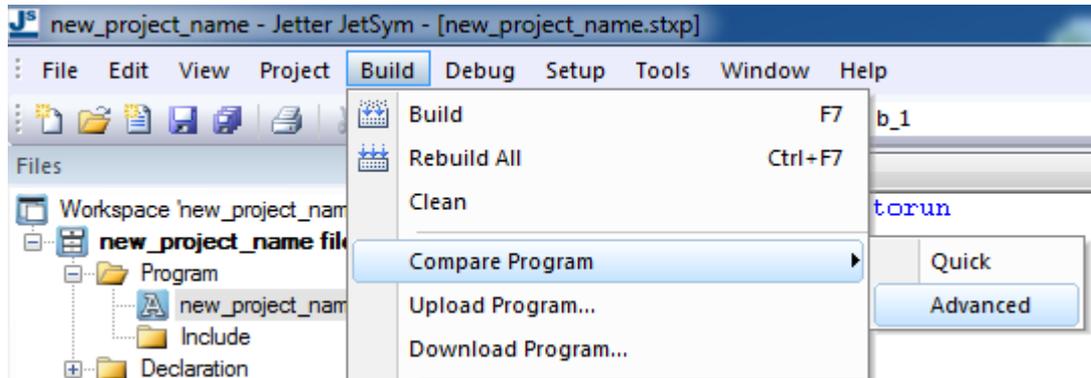
configuration file **config.ini** file.

```
[IP]  
Address      = aaa.bbb.ccc.1  
...
```

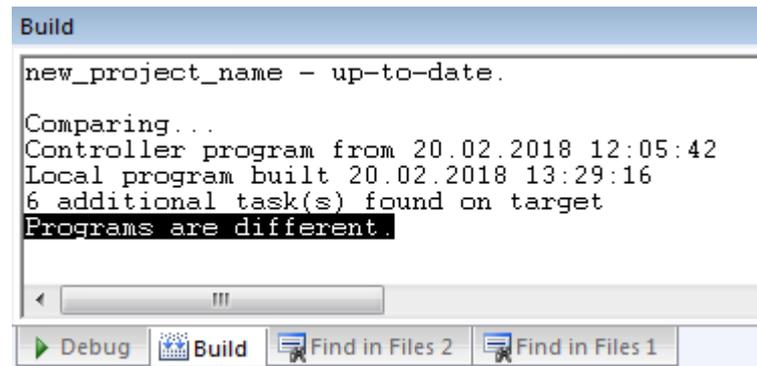


DIP switch								IP address
1	2	3	4	5	6	7	8	
OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	Default IP address
ON	OFF	aaa.bbb.ccc.1						
OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	aaa.bbb.ccc.2
ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	aaa.bbb.ccc.3
...								
OFF	ON	aaa.bbb.ccc.254						
ON	ON	ON	ON	ON	ON	ON	ON	from config.ini

# 项目比较 → 高级



PC项目与控制器当前内部项目的比较!



# 硬件管理

The screenshot displays the hardware management software interface. On the left, a project tree shows a system named 'new\_project\_name System' with various hardware components. The 'JX3-DIO16 (Mod1)' component is selected. The main window shows the configuration tree for this component, with 'Pulse Width Modulation PWM' selected. On the right, a table lists the parameters for this module.

Name	Number	Content	Type	Comment
<b>Pulse Width Modulation PWM</b>				
Enable	515	49152	int	This module n
Prescaler_9_12	526	0	int	The PWM fre
Duty_Out9	527	0	int	Configuration
Duty_Out10	528	0	int	Configuration
Duty_Out11	529	0	int	Configuration
Duty_Out12	530	0	int	Configuration
Prescaler_13_16	531	32768	int	The PWM fre
Duty_Out13	532	0	int	Configuration
Duty_Out14	533	0	int	Configuration
Duty_Out15	534	240	int	Configuration
Duty_Out16	535	15	int	Configuration

The dialog box titled 'bits 100020515' shows the configuration for PWM outputs. The 'Content' field is set to 'PWM for OUT 15 is active, PWM for OUT 16 is'. Below this, there are checkboxes for each output from 9 to 16. The checkboxes for OUT 15 and OUT 16 are checked, while the others are unchecked.

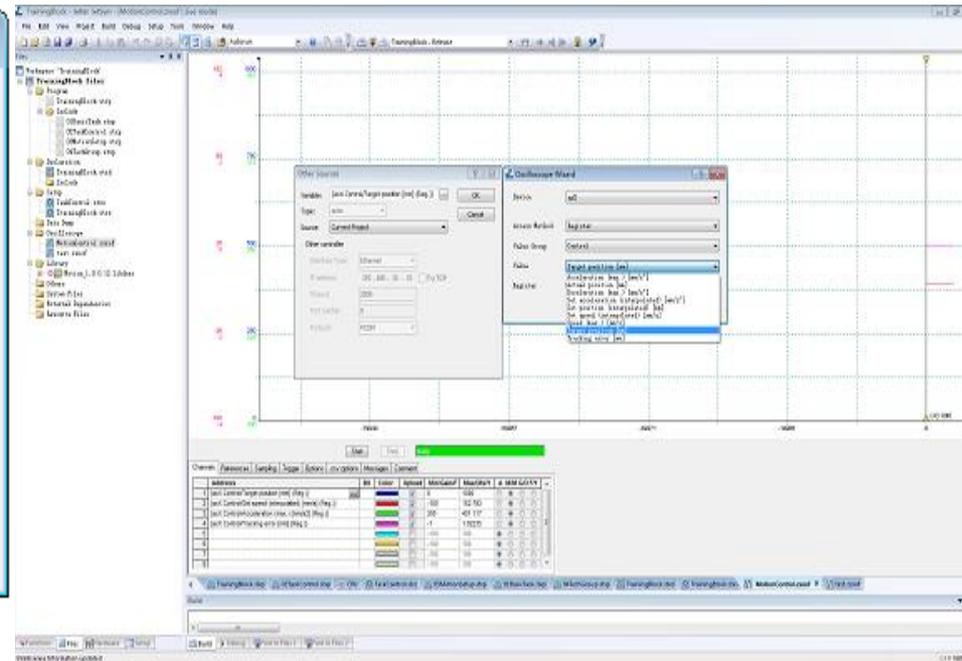
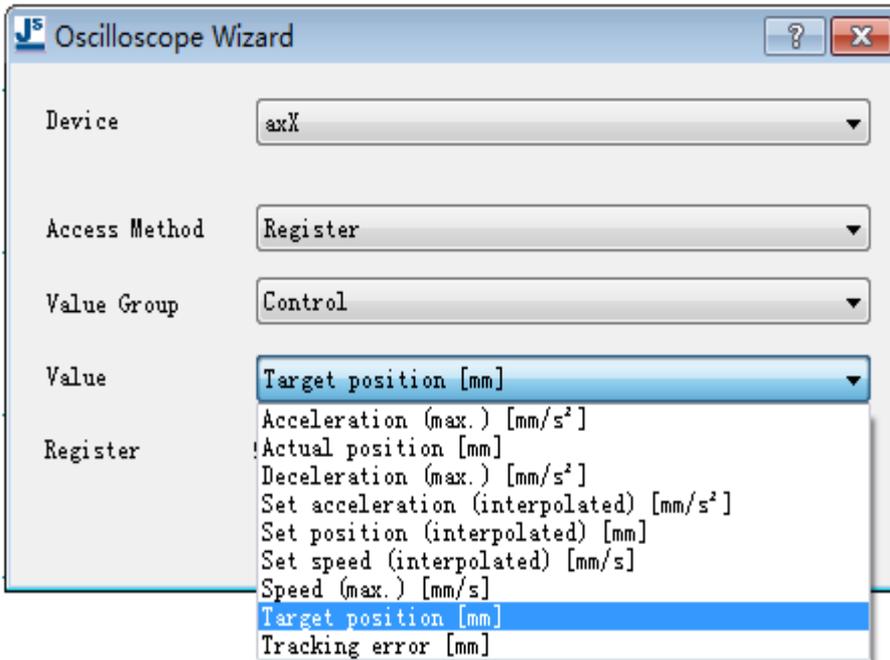
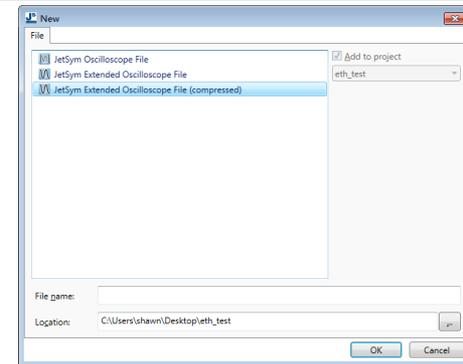
Content: PWM for OUT 15 is active, PWM for OUT 16 is

- PWM for OUT 9 is active
- PWM for OUT 10 is active
- PWM for OUT 11 is active
- PWM for OUT 12 is active
- PWM for OUT 13 is active
- PWM for OUT 14 is active
- PWM for OUT 15 is active
- PWM for OUT 16 is active

# 示波器功能

可以通过JetSym软件自带示波器监控变量，比如：

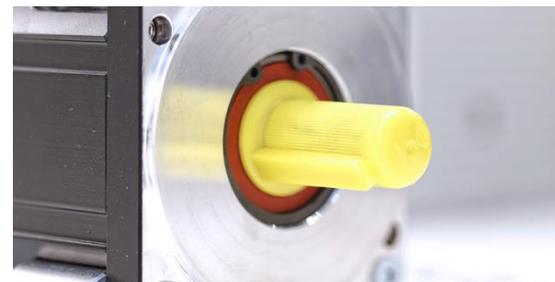
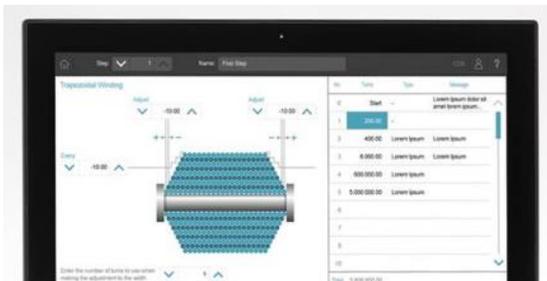
- 开关量输入输出
- 模拟量输入输出
- 伺服轴数据（位置、速度、偏差、加减速等）



# 其他功能

---

- 上传/下载程序（重启）
- 下传/下载变量数据（File.DA）
- 通过JetSym登录文件系统（file system）
- 通过FTP工具登录文件系统（file system）
- 上传/下载源码



# 谢谢!

联系我们:

电话: 021 - 5869 1233 | 邮箱: [contact@jetterat.cn](mailto:contact@jetterat.cn)