

基于 SIMATIC IOT2040 的 Node-red S7 通信

1. 主要目标

基于 Node-red S7 通信，实现 SIMATIC IOT2040 读写 Siemens S7-300/400/1200/1500 或 S7-200 Smart PLC 中的数据。

2. Node-red 及 S7 节点介绍

2.1 Node-red

Node-red 是一种用于以新的有趣的方式将硬件设备、API 和在线服务连接在一起的编程工具。

它提供了一个基于浏览器的编辑器，可以很容易地使用节点栏中的大量节点来组成流，这些节点可以通过一次点击完成部署。

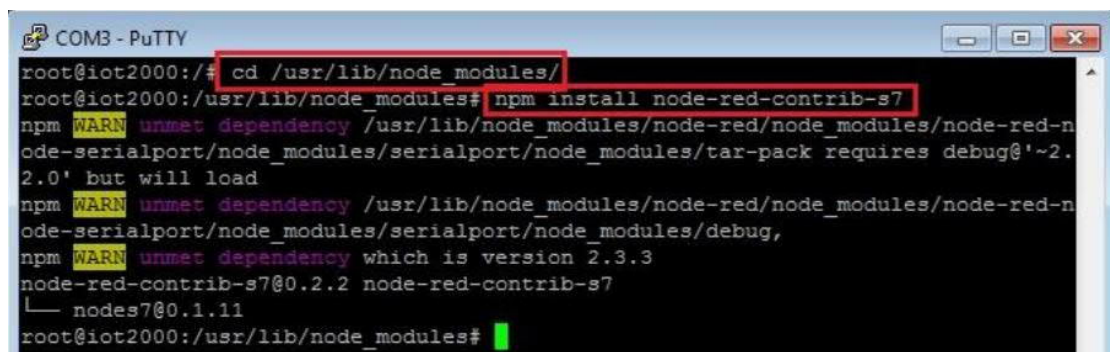
2.2 Node-red S7 节点

S7 节点是一个允许使用西门子 S7 以太网协议 RFC1006 与 S7-300/400/1200/1500 PLC 通信的库。同时，它也适用于 S7-200 Smart PLC。

2.3 安装 Node-red 和 S7 节点

SIMATIC IOT2040 的示例镜像已经预装了 Node-red，而用于 S7 通信的 S7 节点需要用户安装，安装时需要连接网络。

- (1) 打开 putty，建立与 IOT2040 的连接。
- (2) 输入指令 `cd /usr/lib/node_modules`，切换至 Node-red 所在目录。
- (3) 输入指令 `npm install node-red-contrib-s7`，安装 S7 节点。

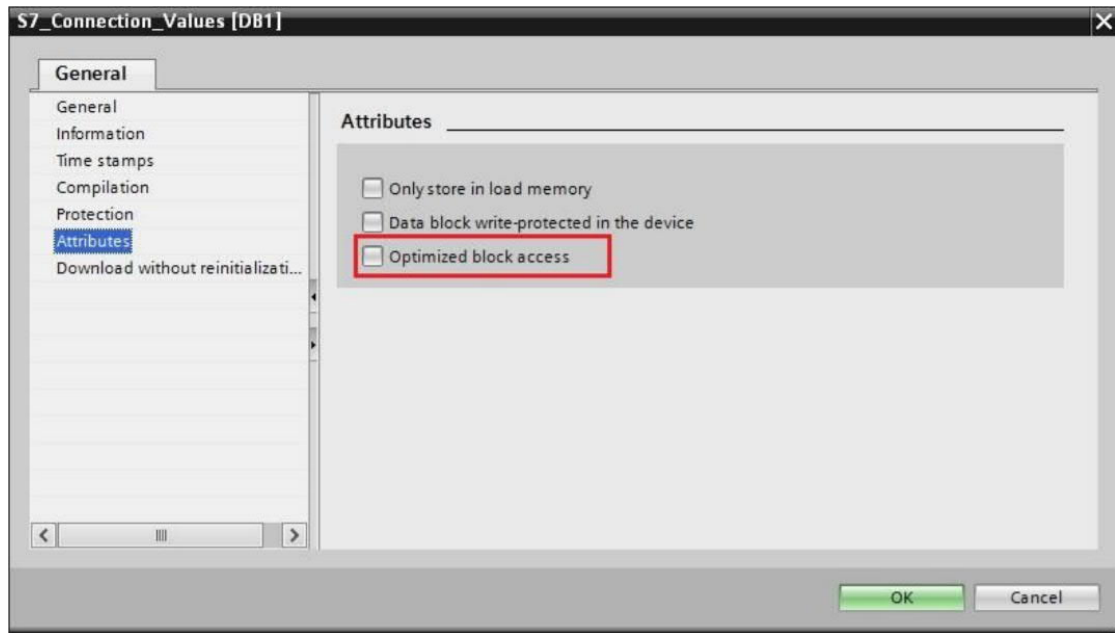


```
COM3 - PuTTY
root@iot2000:/# cd /usr/lib/node_modules/
root@iot2000:/usr/lib/node_modules# npm install node-red-contrib-s7
npm WARN unmet dependency /usr/lib/node_modules/node-red/node_modules/node-red-n
ode-serialport/node_modules/serialport/node_modules/tar-pack requires debug@'~2.
2.0' but will load
npm WARN unmet dependency /usr/lib/node_modules/node-red/node_modules/node-red-n
ode-serialport/node_modules/serialport/node_modules/debug,
npm WARN unmet dependency which is version 2.3.3
node-red-contrib-s7@0.2.2 node-red-contrib-s7
└─ nodes7@0.1.11
root@iot2000:/usr/lib/node_modules#
```

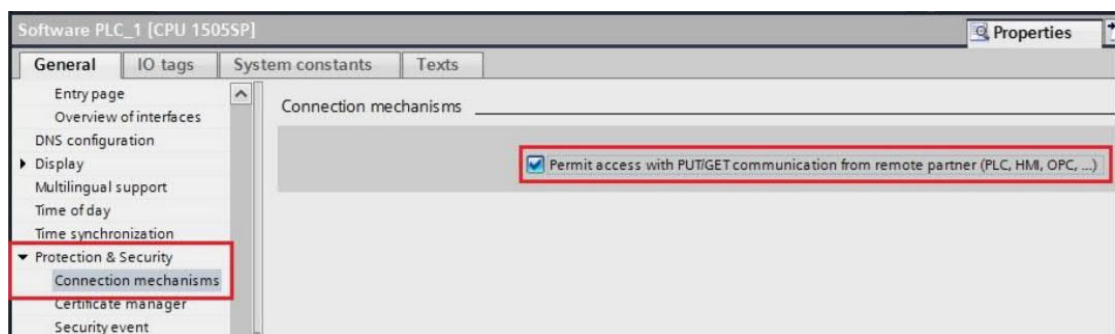
3. 编辑 Node-red S7 通信程序

3.1 S7 通信准备工作

- (1) 只可以使用没有优化的数据块，可以通过 TIA-Portal 在数据块的属性中进行设置。请确保**没有**勾选该选项。



- (2) 必须在 S7-300/400/1200/1500 PLC CPU 配置中启用 PUT/GET 通信，请确保勾选该选项。



- (3) 在 putty 中输入指令 `node /usr/lib/node_modules/node-red/red &`，启动 Node-red。

```
COM3 - PuTTY
root@iot2000:/# node /usr/lib/node_modules/node-red/red &
[1] 1879
root@iot2000:/#

Welcome to Node-RED
=====

20 Dec 09:31:27 - [info] Node-RED version: v0.14.6
20 Dec 09:31:27 - [info] Node.js version: v4.2.4
20 Dec 09:31:27 - [info] Linux 4.4.13-yocto-standard ia32 LE
20 Dec 09:31:27 - [info] Loading palette nodes
20 Dec 09:32:26 - [info] Dashboard version 2.1.0 started at /ui
20 Dec 09:32:28 - [warn] -----
20 Dec 09:32:28 - [warn] [rpi-gpio] Info : Ignoring Raspberry Pi specific node
20 Dec 09:32:28 - [warn] [serialport] Error: Could not locate the bindings file.
Tried:
- /usr/lib/node_modules/node-red/node_modules/node-red-node-serialport/node_modules/serialport/build/serialport.node
- /usr/lib/node_modules/node-red/node_modules/node-red-node-serialport/node_modules/serialport/build/Debug/serialport.node
- /usr/lib/node_modules/node-red/node_modules/node-red-node-serialport/node_modules/serialport/build/Release/serialport.node
- /usr/lib/node_modules/node-red/node_modules/node-red-node-serialport/node_modules/serialport/out/Debug/serialport.node
- /usr/lib/node_modules/node-red/node_modules/node-red-node-serialport/node_modules/serialport/Debug/serialport.node
- /usr/lib/node_modules/node-red/node_modules/node-red-node-serialport/node_modules/serialport/out/Release/serialport.node
- /usr/lib/node_modules/node-red/node_modules/node-red-node-serialport/node_modules/serialport/Release/serialport.node
- /usr/lib/node_modules/node-red/node_modules/node-red-node-serialport/node_modules/serialport/build/default/serialport.node
- /usr/lib/node_modules/node-red/node_modules/node-red-node-serialport/node_modules/serialport/compiled/4.2.4/linux/ia32/serialport.node
20 Dec 09:32:28 - [warn] -----
20 Dec 09:32:28 - [info] Settings file : /home/root/.node-red/settings.js
20 Dec 09:32:28 - [info] User directory : /home/root/.node-red
20 Dec 09:32:28 - [info] Flows file : /home/root/.node-red/flows_iot2000.json
20 Dec 09:32:28 - [info] Creating new flow file
20 Dec 09:32:28 - [info] Starting flows
20 Dec 09:32:28 - [info] Started flows
20 Dec 09:32:28 - [info] Server now running at http://127.0.0.1:1880/

root@iot2000:/#
```

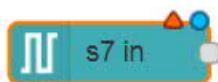
(4) 在浏览器中打开 IOT2040 IP 地址的 1880 端口 (例如 <http://192.168.200.1:1880>)。



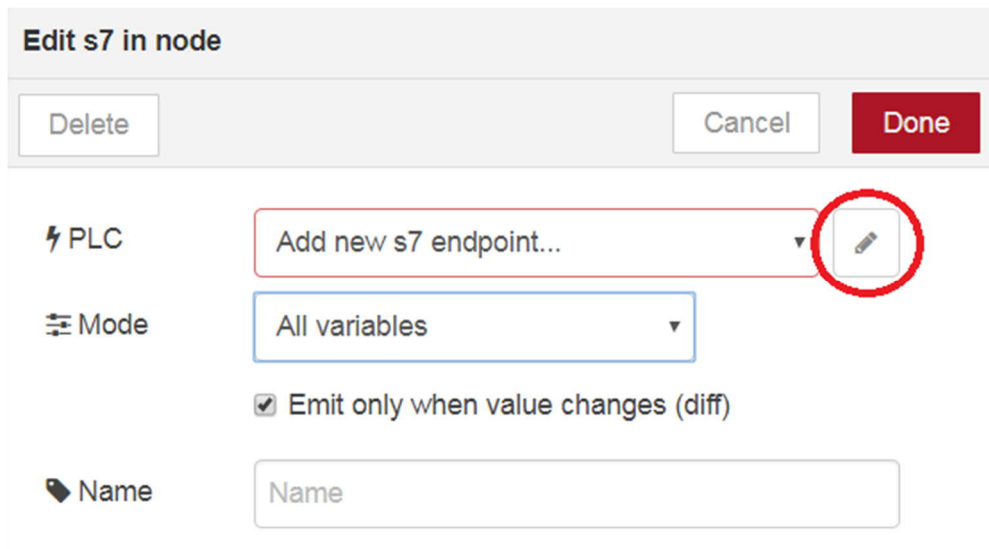
3.2 编辑 S7 in 节点

利用“s7 in”节点可以基于 S7 协议从 S7 PLC 读取数据。

(1) 在左侧节点栏中选择“s7 in”节点，拖动添加至编辑区域。

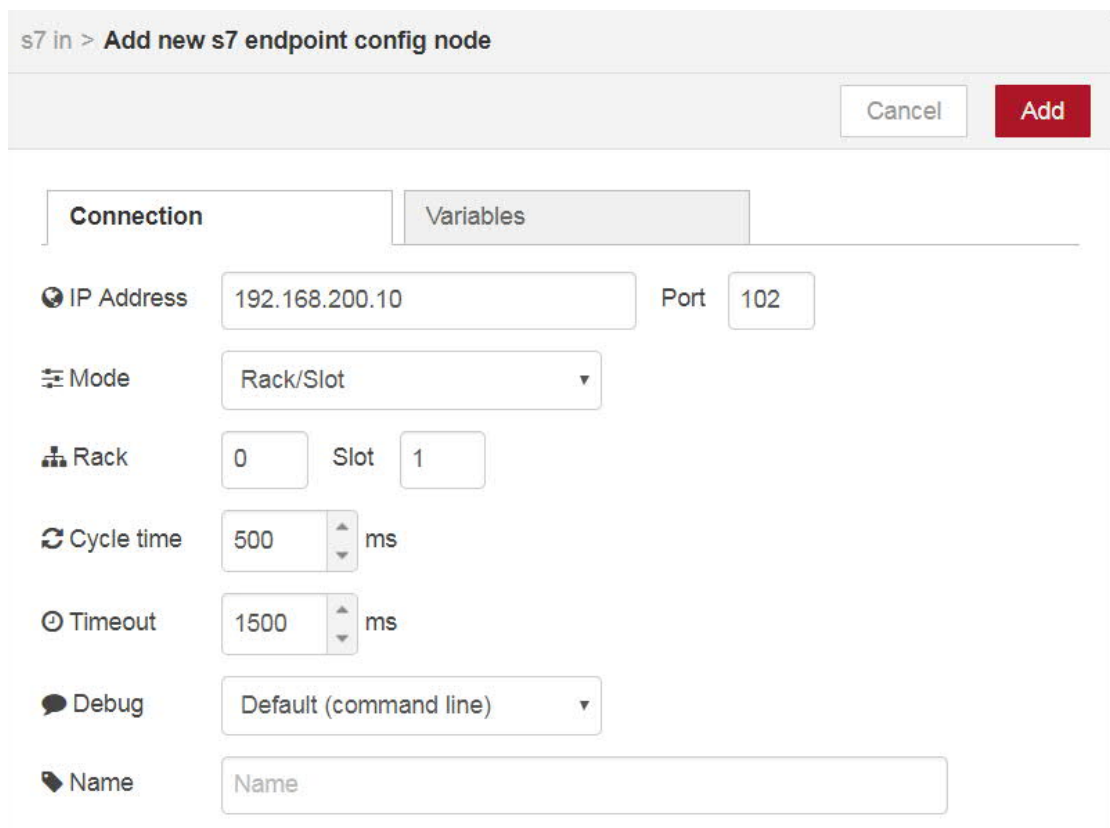


(2) 双击“s7 in” 节点。将 Mode 选为 All variables。点击图中图标编辑 S7 端点。



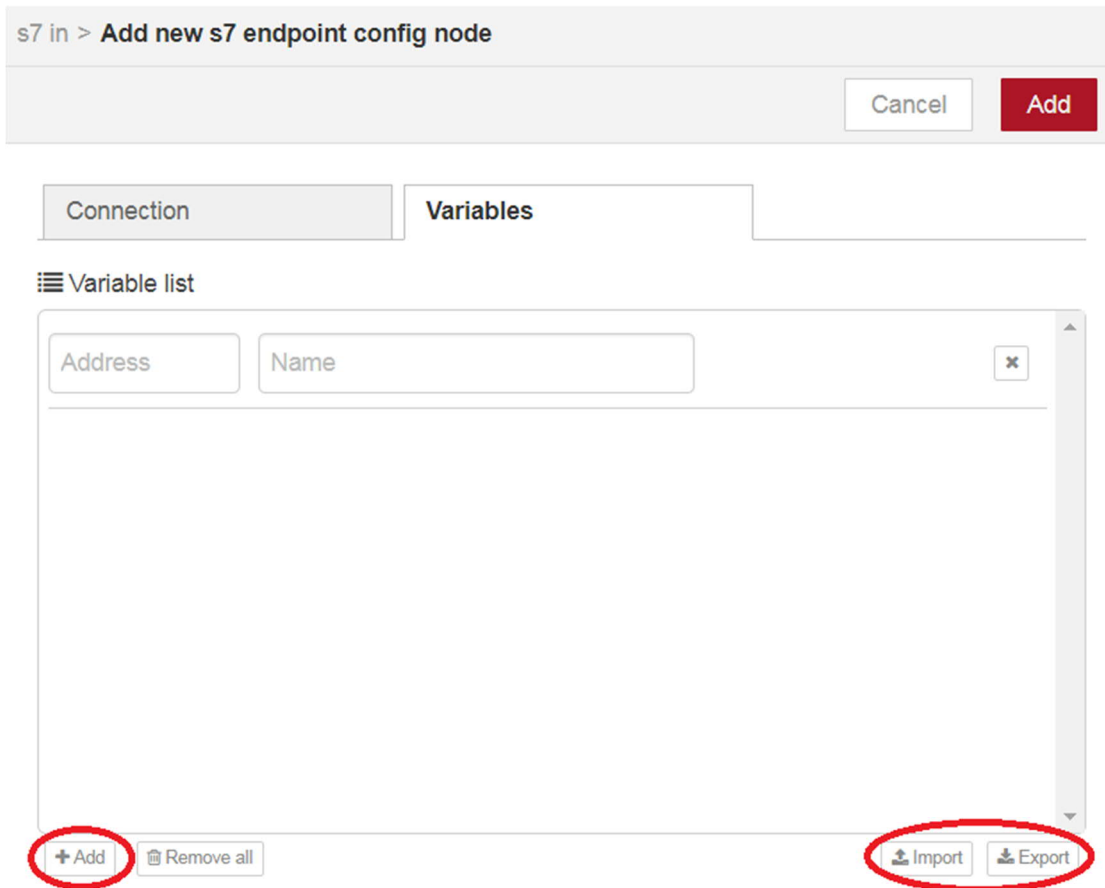
(3) 配置 S7 端点的连接信息。

S7 端点的连接信息包括 S7 PLC 的 IP 地址、端口号、机架号、槽号、读取周期等信息。其中，默认的端口号是 102。不同的 S7 PLC，槽号也不同。



(4) 配置 S7 端点的变量信息列表。

S7 端点的变量信息包括变量的寻址方式以及变量名称。使用“+Add”按钮来添加新的变量，“Export”按钮可以将变量列表导出至 .csv 文件中，“Import”按钮可以通过 .csv 文件导入变量列表。



(5) 示例

STEP 7-MicroWIN Smart 中 DB 块数据（以 S7-200 Smart 为例）：



TIA Portal 中 DB 块数据（以 S7-1500 为例）：

项目1 > PLC_1 [CPU 1511-1 PN] > 程序块 > 数据块_1 [DB1]

数据块_1									
	名称	数据类型	偏移量	启动值	保持性	可从 HMI ...	在 HMI ...	设置值	注释
1	Static				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
2	Temperature	Real	0.0	20.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	Auto Count	Byte	4.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4	Check	Bool	5.0	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
5	Char	Char	6.0	'c'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
6	String	String	8.0	'abc'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Node-red 中对应的部分变量信息列表：

☰ Variable list

Q0.1	Q
MB0	MB
DB1,B4	AC
DB1,C6	CHAR
DB1,S7.3	STRING

3.3 编辑 S7 out 节点

利用“s7 out”节点可以基于 S7 协议向 S7 PLC 写入数据。

- (1) 在左侧节点栏中选择“s7 out”节点，拖动添加至编辑区域。



- (2) 双击“s7 out”节点，编辑节点。

Edit s7 out node

⚡ PLC

🔄 Variable

📁 Name

Caution when writing data to production PLCs!

(3) 如“s7 in”节点已添加过 S7 端点，则直接选择相应 S7 端点即可；如未添加过，请参考“s7 in”节点添加 S7 端点过程进行添加。

Edit s7 out node

Delete Cancel Done

⚡ PLC Add new s7 endpoint... 192.168.2.1:102:0:1 192.168.200.10:102:0:1 Add new s7 endpoint...

🔄 Variable

📁 Name Name

Caution when writing data to production PLCs!

(4) 通过下拉菜单选择要写入的变量，每个“s7 out”节点只能对一个变量进行写入操作。

Edit s7 out node

Delete Cancel Done

⚡ PLC 192.168.200.10:102:0:1

🔄 Variable Select a variable Select a variable Q MB AC CHAR STRING

📁 Name

Caution when v

在选择完成后，点击“Done”按钮确定。

Edit s7 out node

Delete
Cancel
Done

⚡ PLC

192.168.200.10:102:0:1

✎

🔄 Variable

Q

▼

Q0.1

📁 Name

Name

Caution when writing data to production PLCs!

说明：

- 1) 不可对 I 区进行写入
- 2) 若 M 点及 Q 点在 PLC 程序中为被写入对象，则不可对 M 点及 Q 点进行写入
- 3) 待写入的变量只可从已输入的变量列表中选择

4. S7 PLC 数据寻址方式及 Node-red 对应寻址方式

下表给出了 S7 PLC 中数据寻址方式及 Node-red 对应寻址方式。

由于 CPU 存储方式的不同及 Node-red S7 节点限制，有些数据无法直接读写，但是可以将它们转化为其他数据区域中的变量进行读写。

4.1 S7-300/400/1200/1500

	数据区域	数据类型	PLC 寻址方式	Node-red 寻址方式
PLC 变量	I	BOOL	Ix.y	Ix.y
		BYTE,CHAR	IBx	IBx
		WORD,INT	IWx	IWx
		DWORD,DINT, REAL	IDx	IDx
	Q	BOOL	Qx.y	Qx.y
		BYTE,CHAR	QBx	QBx
		WORD,INT	QWx	QWx

	M	DWORD,DINT, REAL	QD _x	QD _x
		BOOL	M _{x.y}	M _{x.y}
		BYTE,CHAR	MB _x	MB _x
		WORD,INT	MW _x	MW _x
	DWORD,DINT, ,REAL	MD _x	MD _x	
	T	TIMER (only support S7-300/400)	T _n	
C	COUNTER (only support S7-300/400)	C _n		
用户数据块	DB	REAL		DB _n ,REAL _x
		DWORD		DB _n ,DWORD _x
		DINT		DB _n ,DINT _x
		INT		DB _n ,INT _x
		WORD		DB _n ,WORD _x
		BYTE or B		DB _n ,B _x
		STRING or S		DB _n ,S (x -1).length
		CHAR or C		DB _n ,C _x

说明：

- 1) n 为数据区域偏移量
- 2) x 为数据字节偏移量
- 3) y 为数据位偏移量
- 4) 确保“STRING”或“S”的数据字节偏移量非 0，length 替换为字符串长度

4.2 S7-200 Smart (以 SR60 为例)

	数据区域	数据类型	PLC 寻址方式	Node-red 寻址方式
PLC 变量	I	BOOL	Ix.y	Ix.y
		BYTE	IBx	IBx
		WORD,INT	IWx	IWx
		DWORD,DINT	IDx	IDx
	Q	BOOL	Qx.y	Qx.y
		BYTE	QBx	QBx
		WORD,INT	QWx	QWx
		DWORD,DINT	QDx	QDx
	M	BOOL	Mx.y	Mx.y
		BYTE	MBx	MBx
		WORD,INT	MWx	MWx
		DWORD,DINT	MDx	MDx
	T			
	C			
	SM			
	HC			
L				
S				
用户数据块	DB(V)	DWORD	DBn VDx	DBn,DWORDx
		DINT	DBn VDx	DBn,DINTx
		INT	DBn VWx	DBn,INTx
		WORD	DBn VWx	DBn,WORDx
		STRING or S	DBn VBx	DBn,B(x-1).length
		BYTE or B	DBn VBx	DBn,Bx

说明:

- 1) n 为数据区域偏移量
- 2) x 为数据字节偏移量
- 3) y 为数据位偏移量
- 4) 确保“STRING”或“S”的数据字节偏移量非 0, length 替换为字符串长度