

Propojení dvou UniPi pomocí SSCP

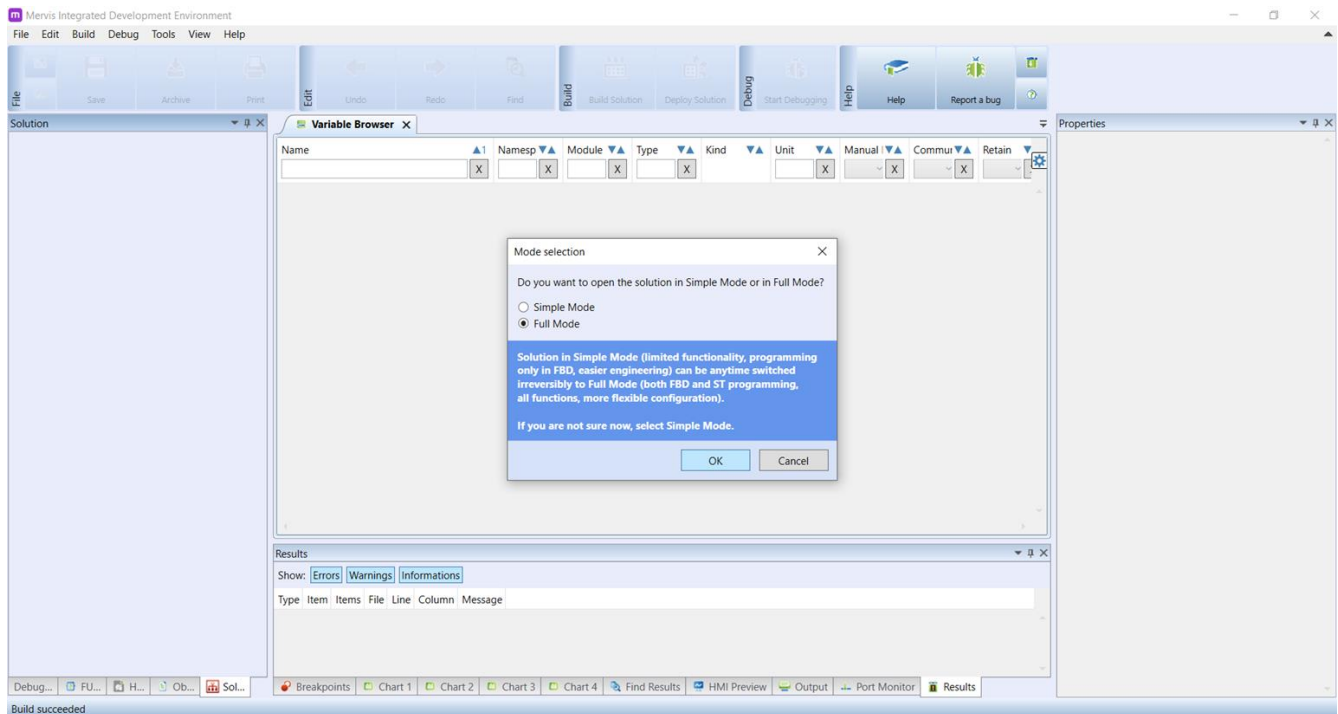
Pokud potřebujete z jednoho UniPi číst vstupy, nebo zapisovat na výstupy jiného UniPi, můžete využít protokolu SSCP. V komplexnějším řešení můžete SSCP využít i na předávání výstupů z čidel, nebo k předávání proměnných. V následujícím tutoriálu vás naučíme, jak protokol SSCP nastavit.

Co budeme potřebovat?

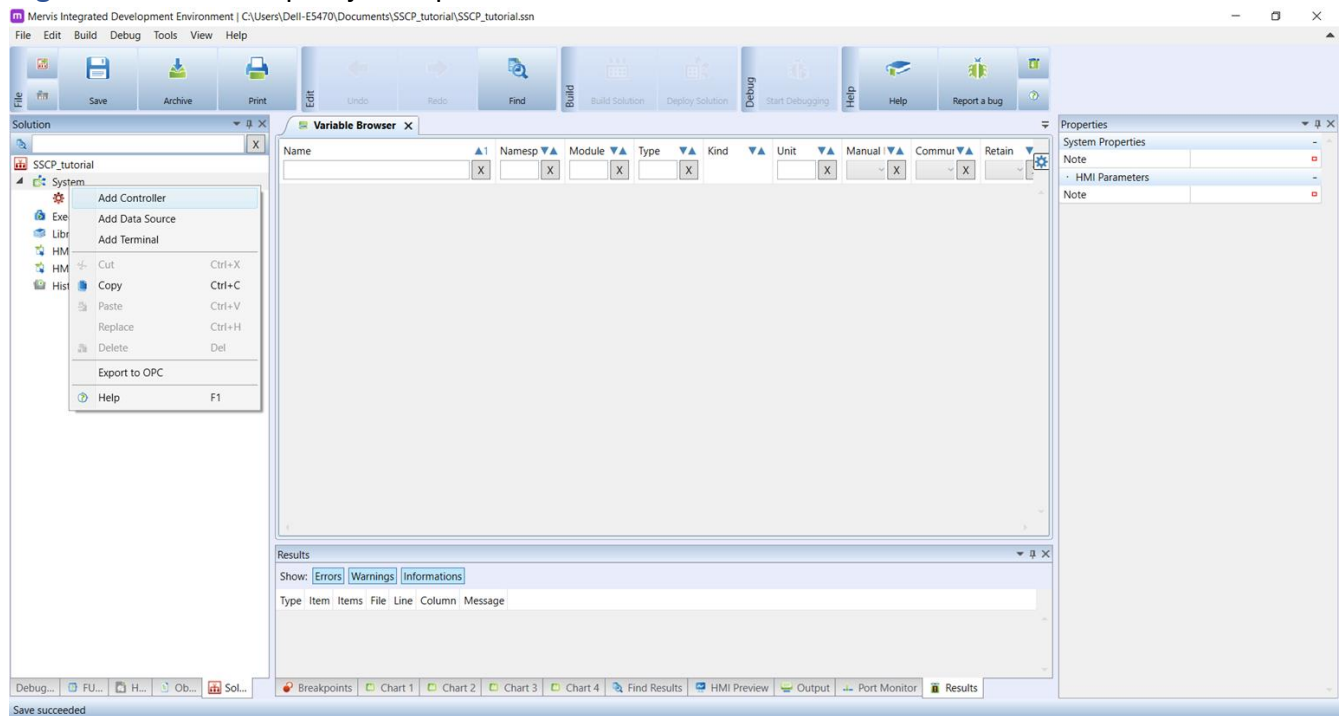
- Dvě jednotky UniPi s Mervis OS
- 24V DC napájení
- Připojení k místní síti (libovolný switch, nebo router)
- 3ks síťového kabelu (RJ45)
- **Pro pokročilé:** 2ks 1-Wire teplotní čidlo (Attaching 1-Wire devices)

1 Základní nastavení projektu v Mervisu

Vytvoříme nový projekt v plném módu.

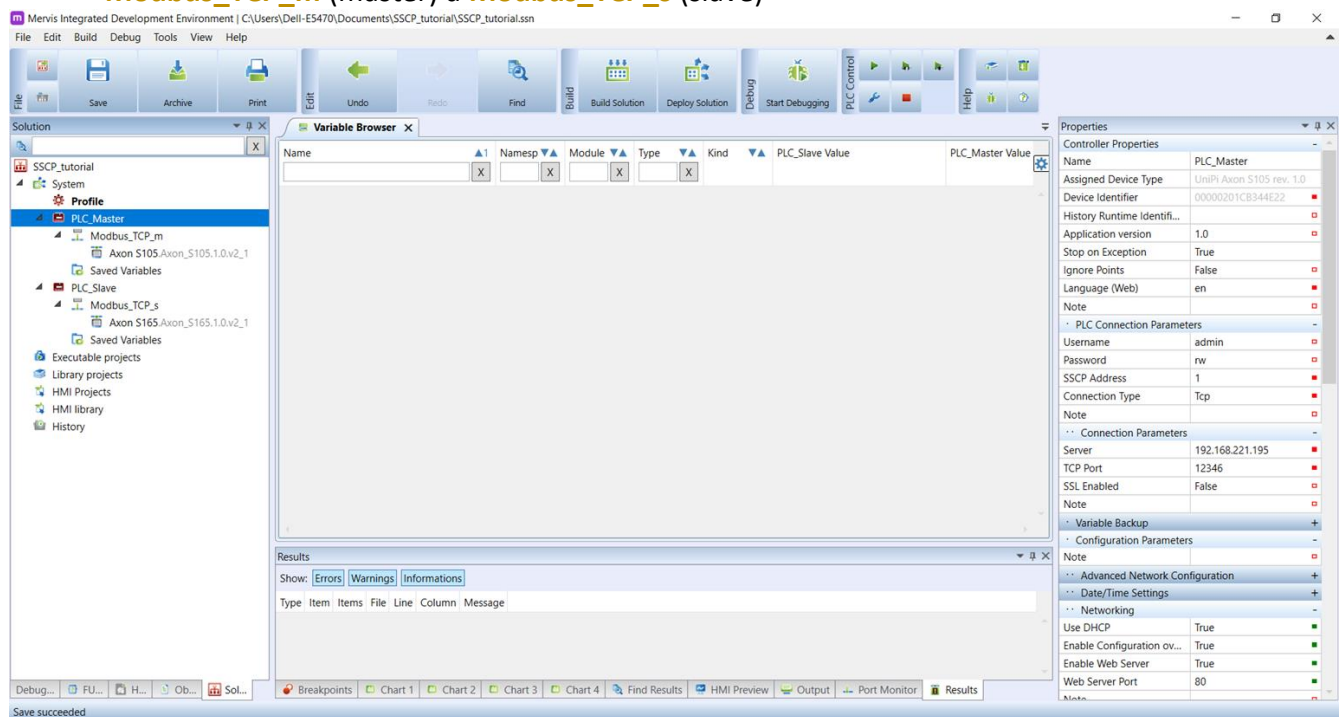


Nyní si přidáme jednotlivá PLC tak, že pravým tlačítkem myši klikneme na **system** a zvolíme **přidat regulátor**. Tuto akci opakujeme i pro druhé PLC.

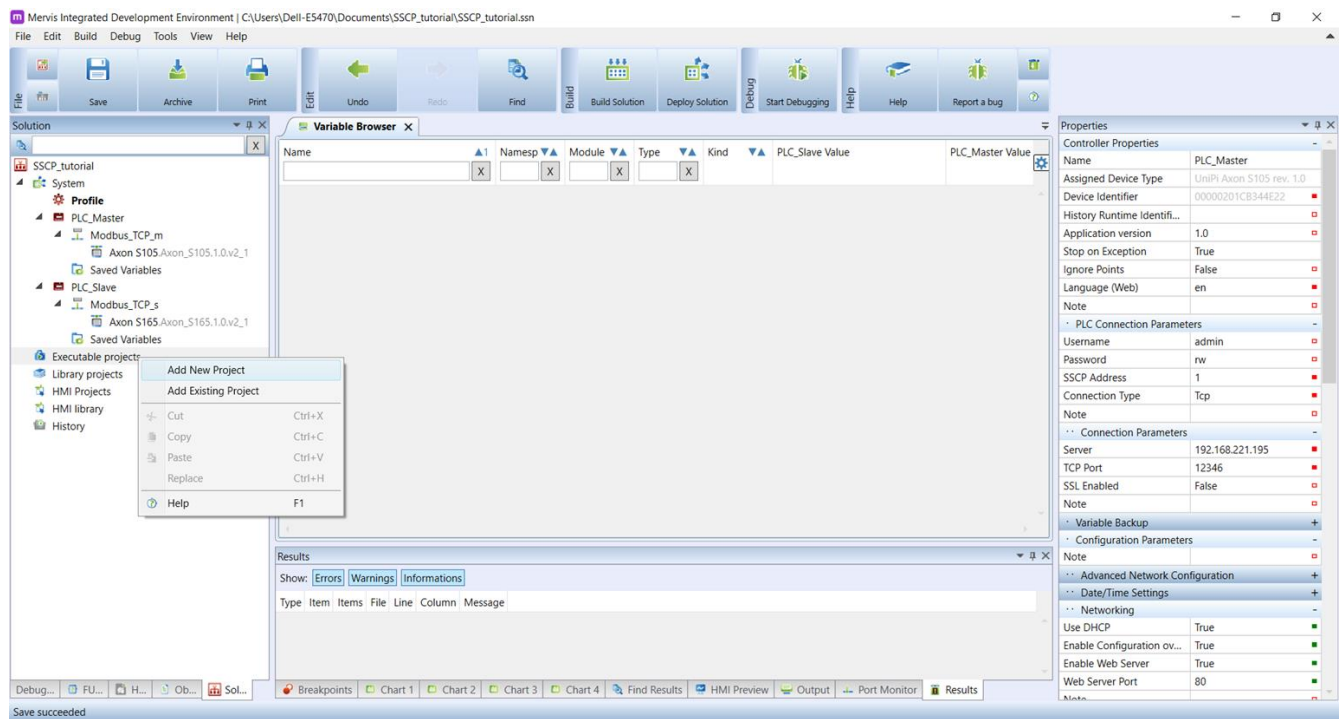


Pro přehlednost doporučujeme vše pojmenovat

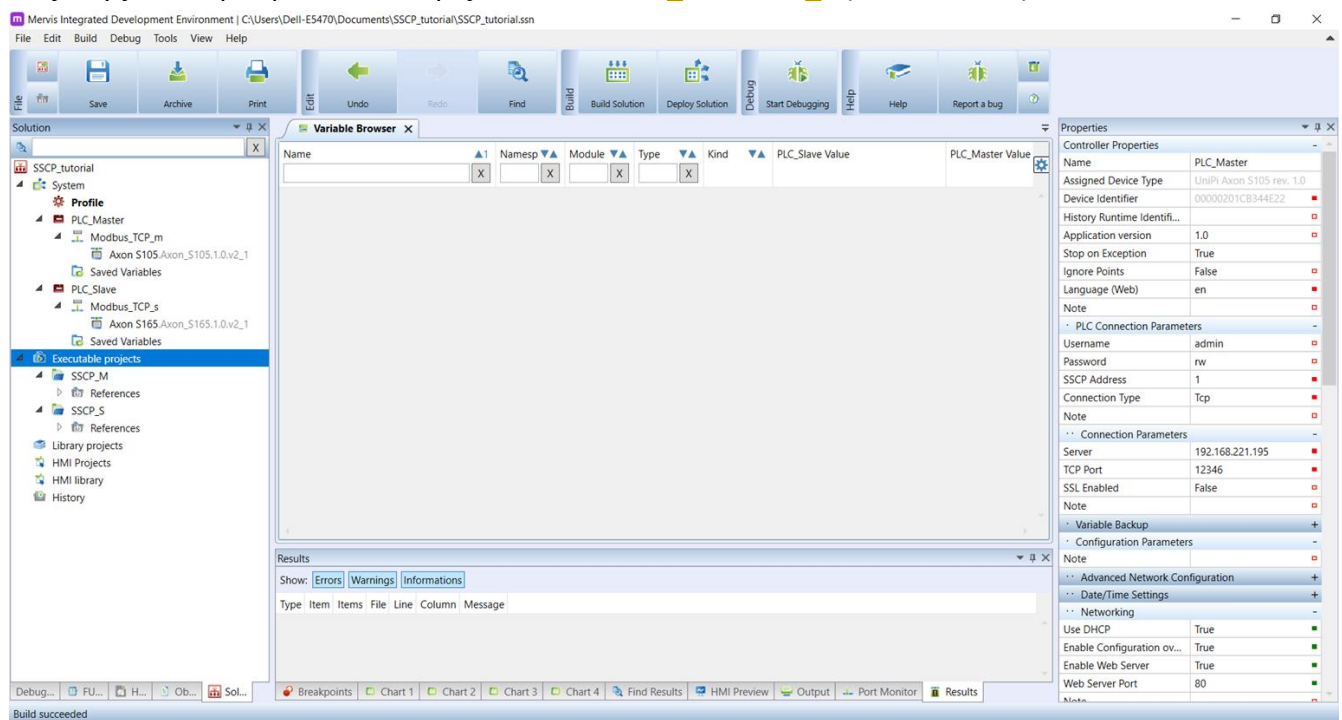
- První UniPi jako **PLC_Master**
- Druhé UniPi jako **PLC_Slave**
- Doporučujeme přejmenovat i automaticky vytvořené kanály (channel, channel_0) jako **Modbus_TCP_m** (master) a **Modbus_TCP_s** (slave)



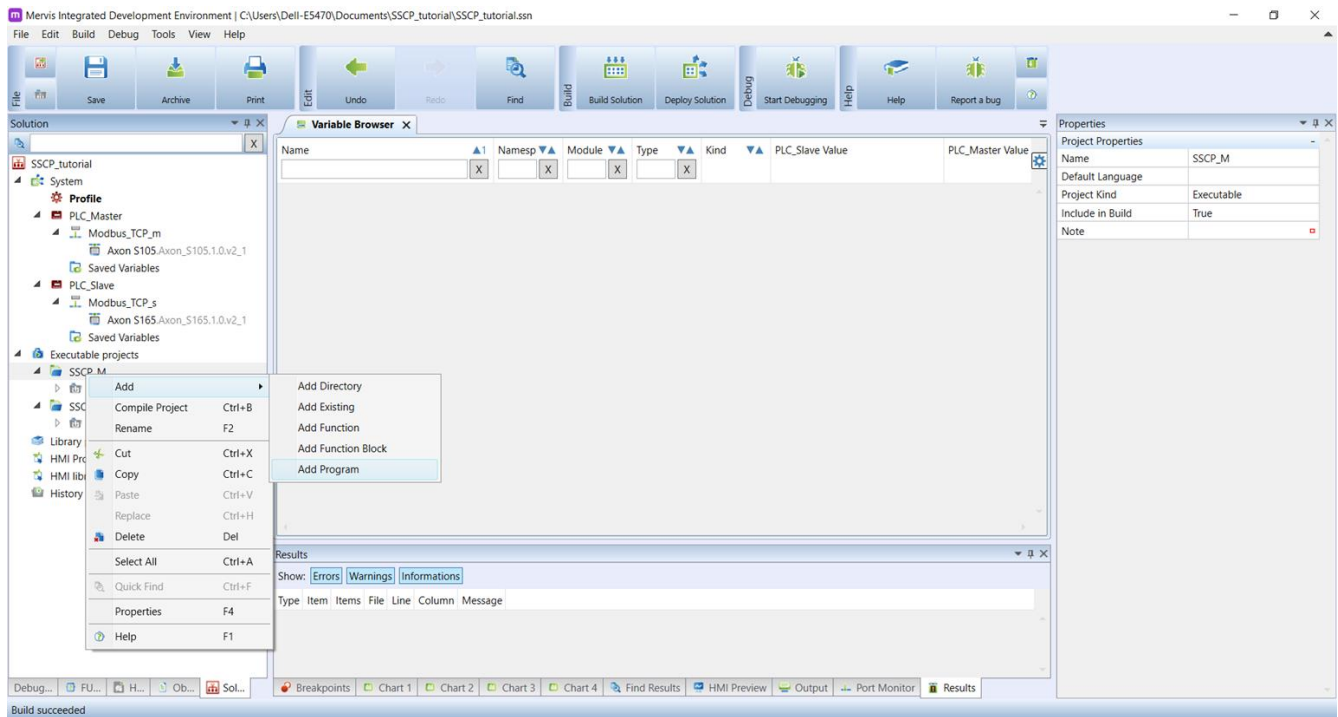
Dále pro každou jednotku vytvoříme vlastní projekt. Klikneme pravým tlačítkem myši na **spustitelné projekty** a vybereme **přidat nový projekt**.



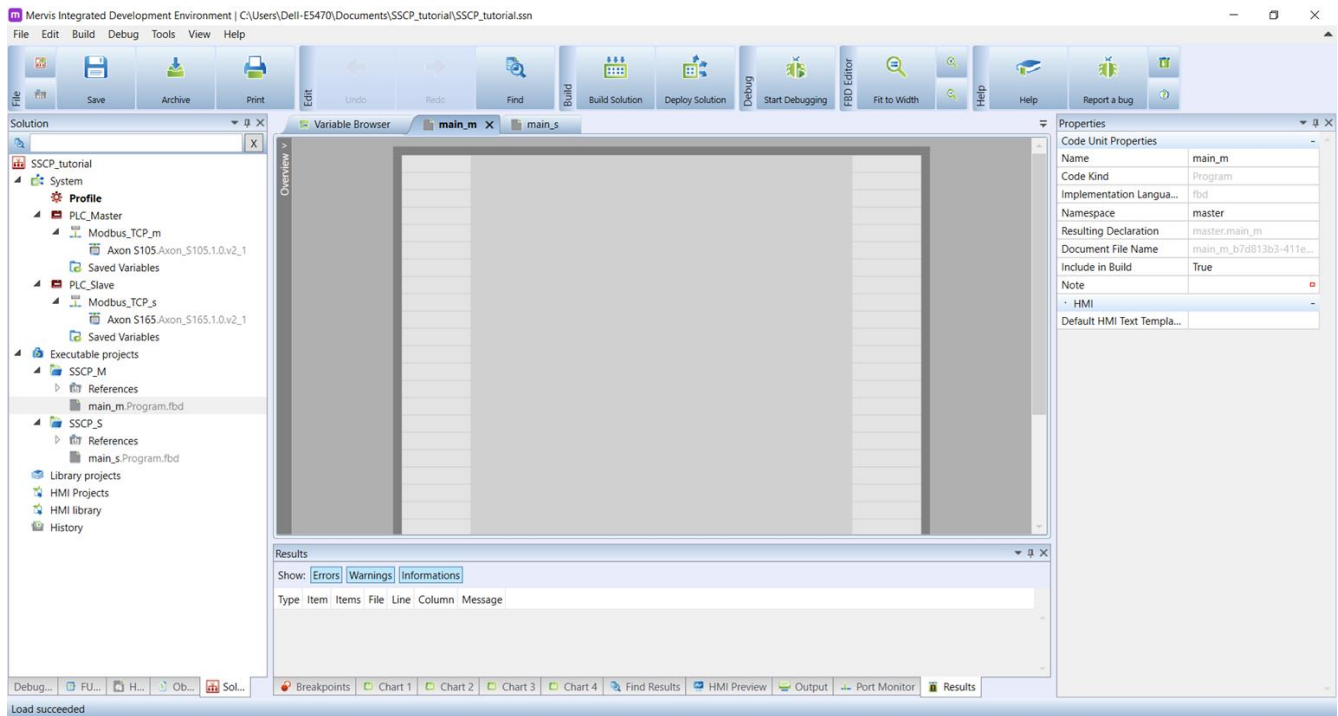
Projekty jsme si pro přehlednost pojmenovali **SSCP_M** a **SSCP_S** (master, slave).



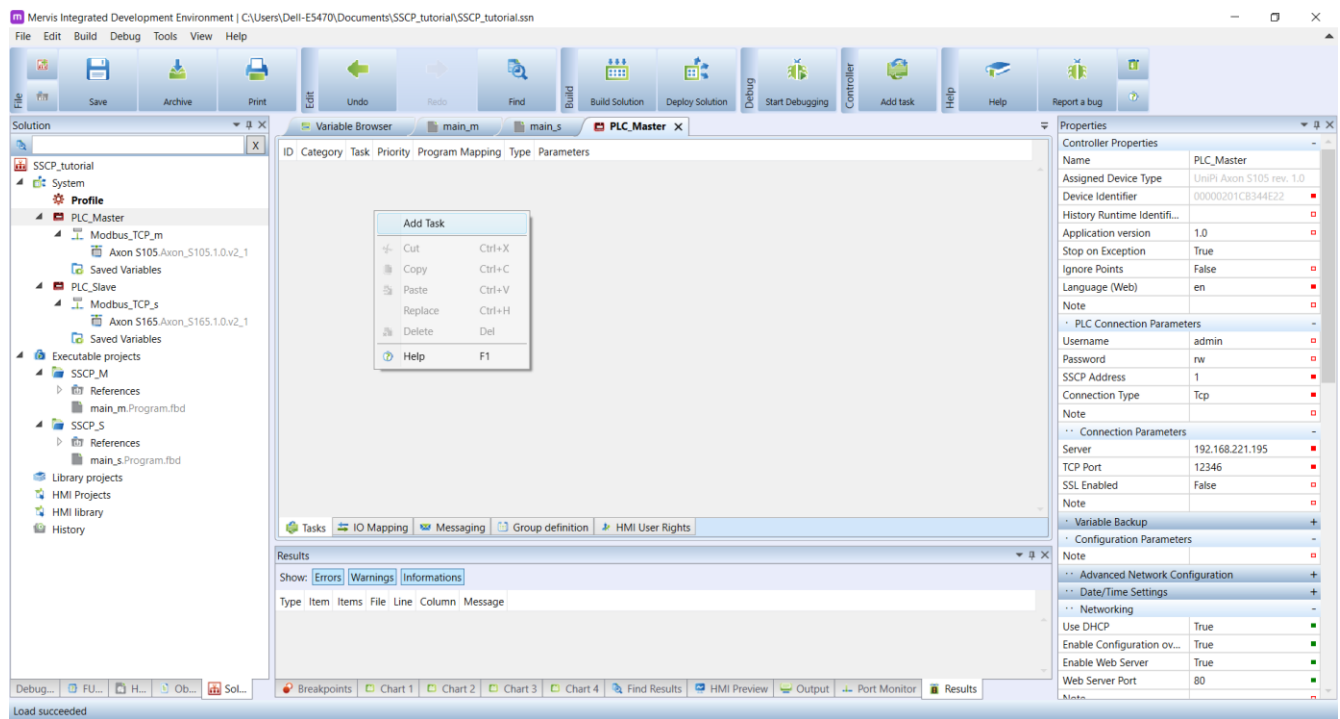
Klikneme pravým tlačítkem myši na projekt (**SSCP_M**) a vybereme **přidat** a **přidat program**. Program nazveme jako **main_m**. Postup opakujeme i pro projekt (**SSCP_S** → **main_s**).



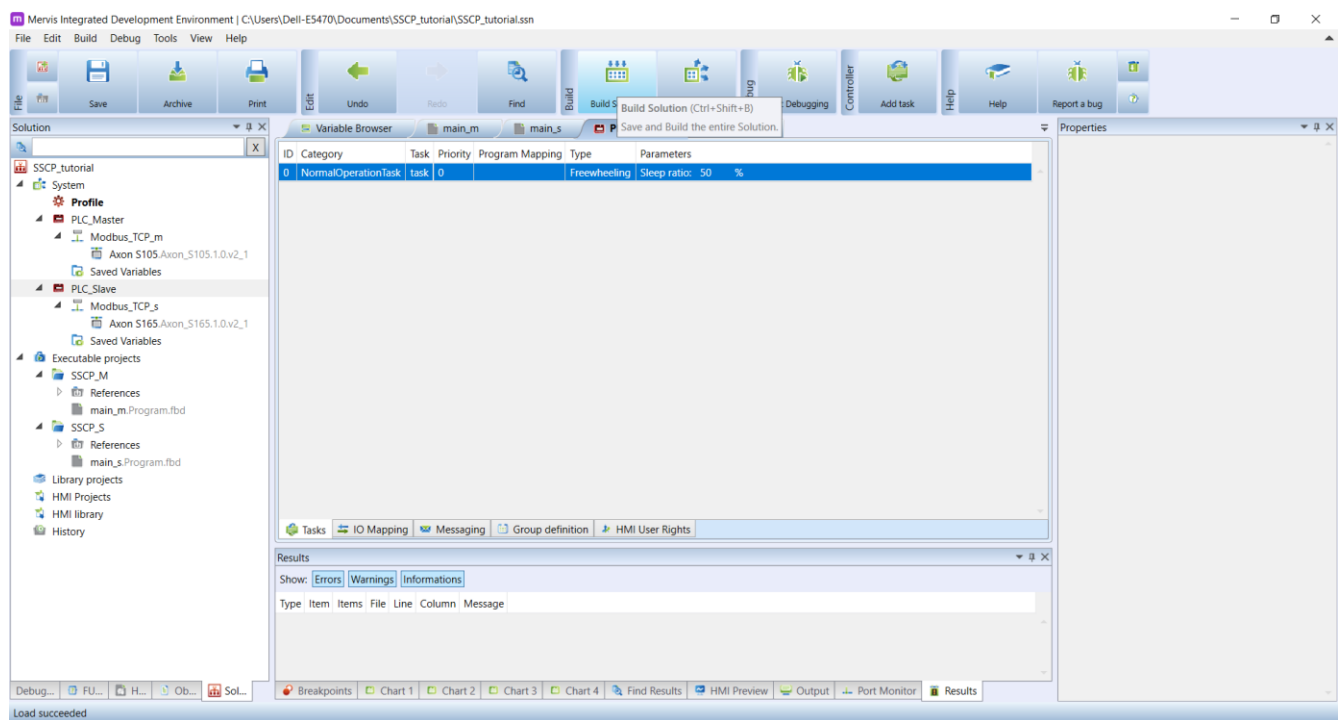
Nyní by vaše nastavení mělo vypadat takto:



Pomocí dvojkliku levým tlačítkem na **PLC_Master** otevřete nabídku PLC. Zde do volného prostoru klikněte pravým tlačítkem a v nabídce vyberte **přidat task**. Následně opakujte pro **PLC_Slave**.

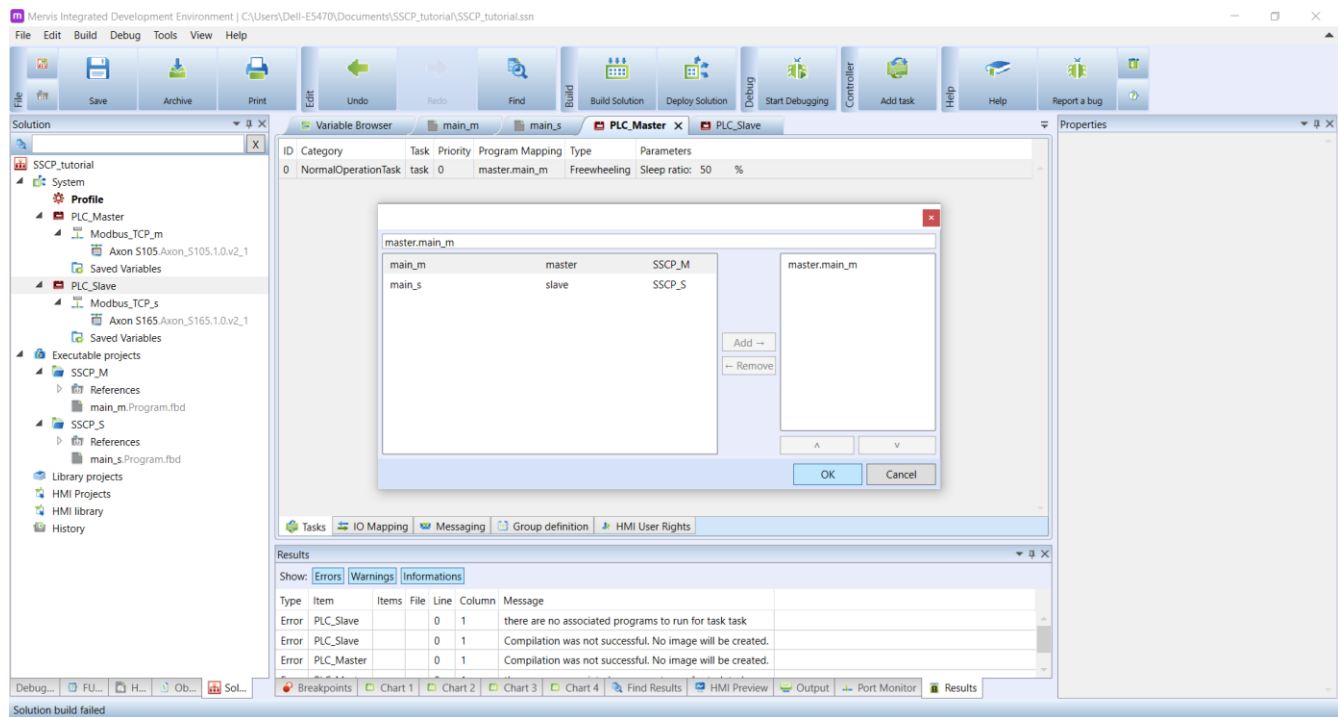


Ted' vše uložte a spusťte kompilaci tlačítkem **zkompilovat sestavu**.



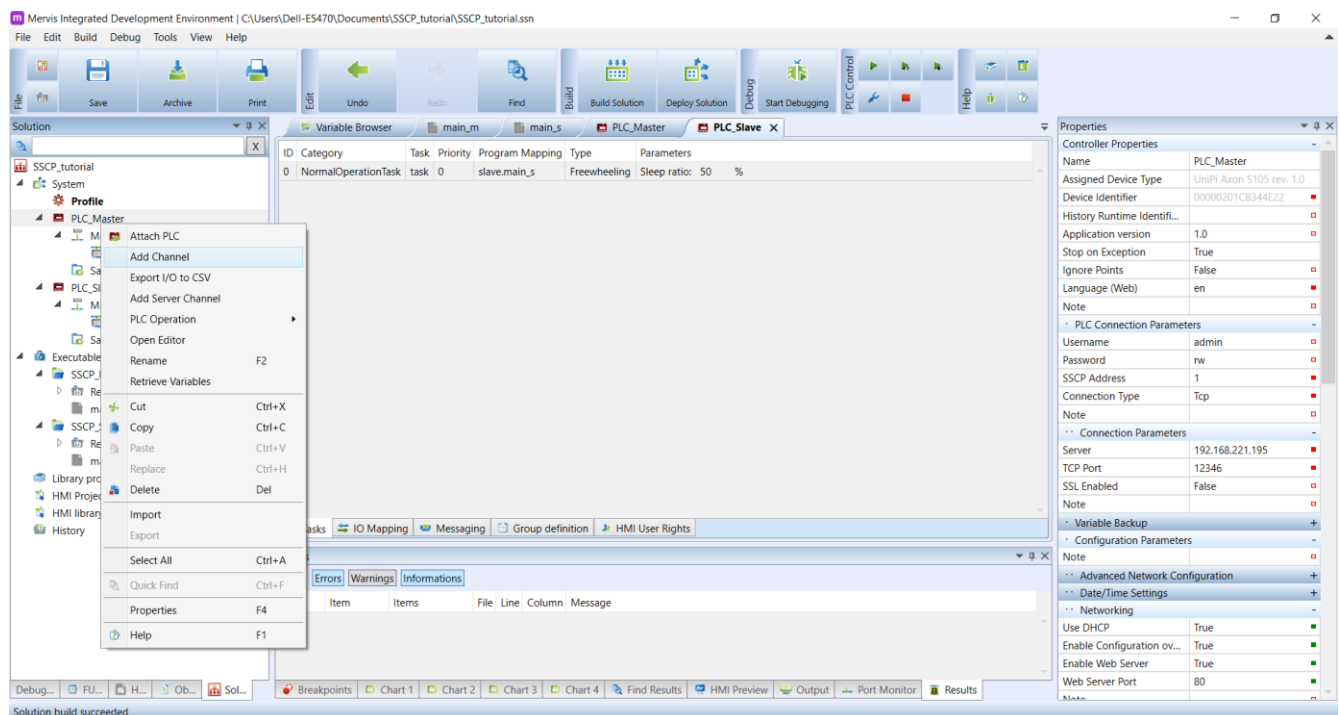
Vypíše se nám chybová hláška, protože projekt není dokončen. Nyní již ale můžete namapovat program do tasku. Dvakrát klikněte na prázdné místo ve sloupci **mapování programu**.

Přidejte **main_m** do **PLC_Master** a **main_s** do **PLC_Slave**.

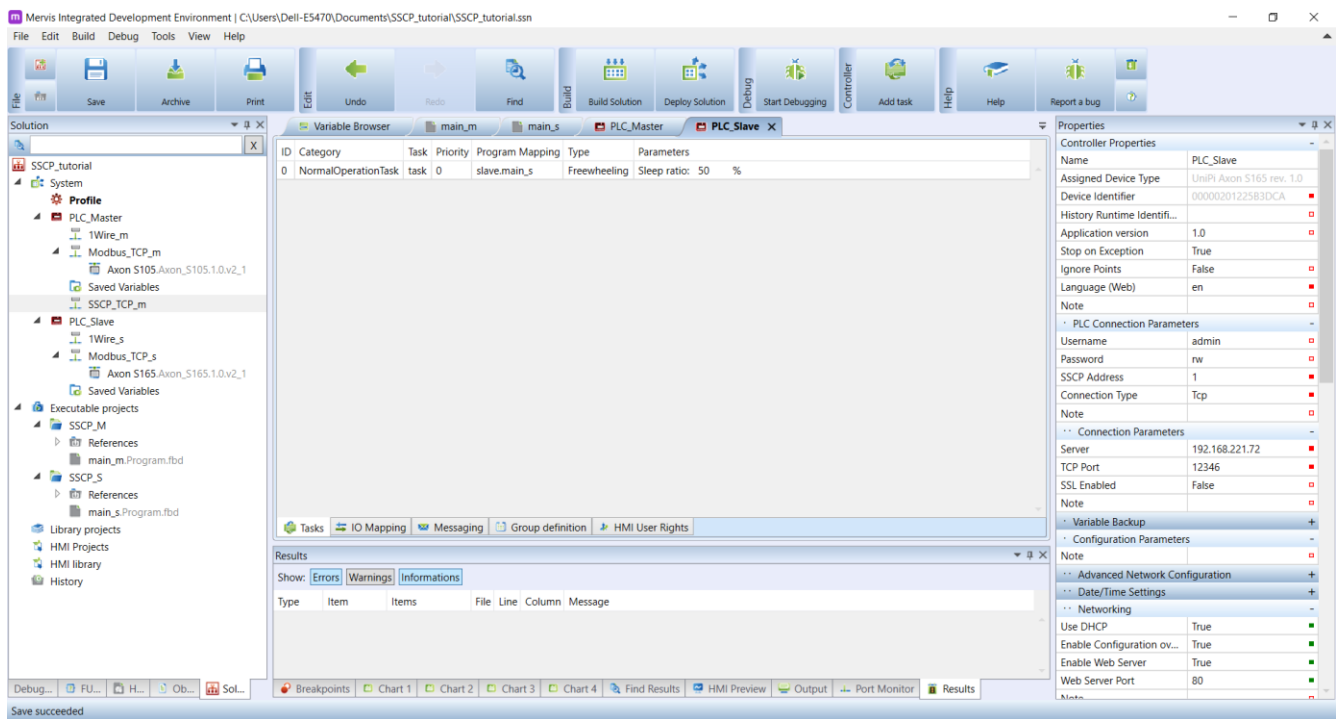


Nyní si můžete přidat další **kanály** podle vašich potřeb. Pro naše účely si vytvoříme **kanál** pro **SSCP_TCP_m** u **PLC_Master**.

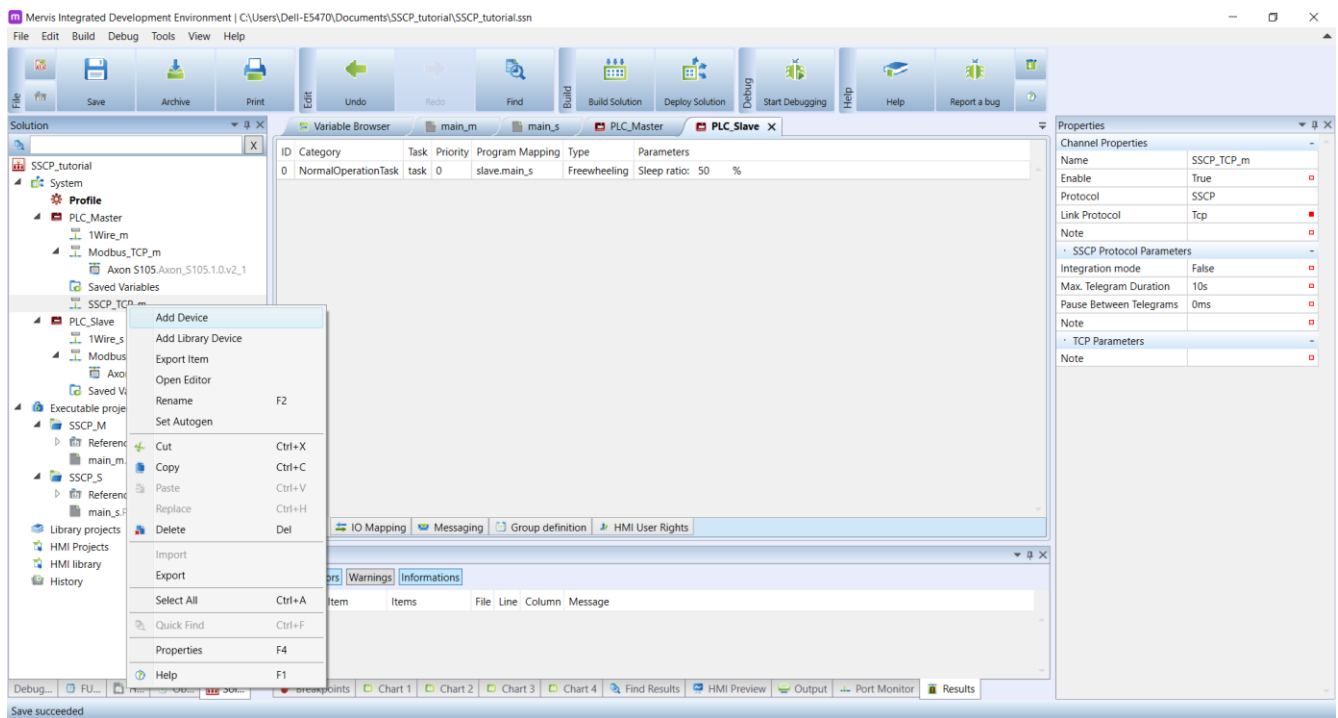
Pro pokročilé: Vytvořte i kanál **1-Wire** sběrnice u každého PLC.



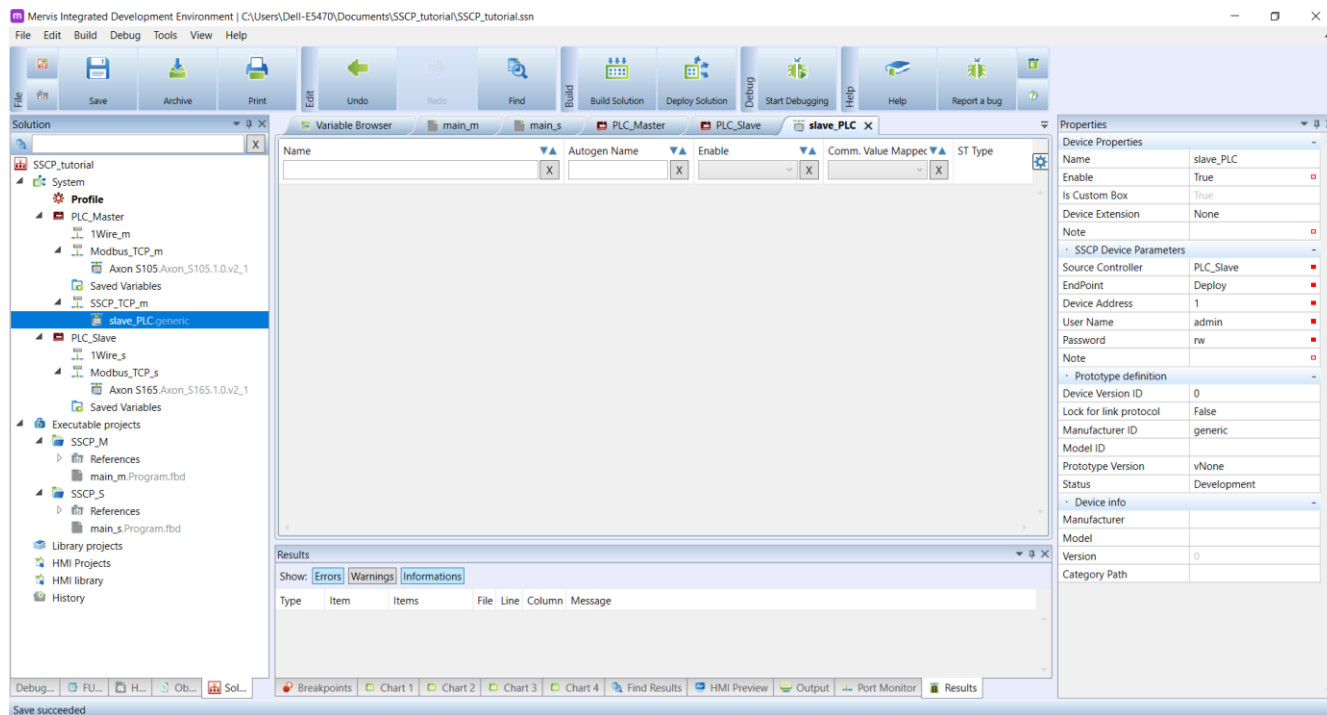
Nezapomeňte vše pojmenovat, rozlišujte **PLC_Master** a **PLC_Slave**. Výsledek by měl vypadat podobně jako na obrázku:



Klikněte pravým tlačítkem myši na kanál **SSCP_TCP_m** a vyberte **přidat zařízení**.

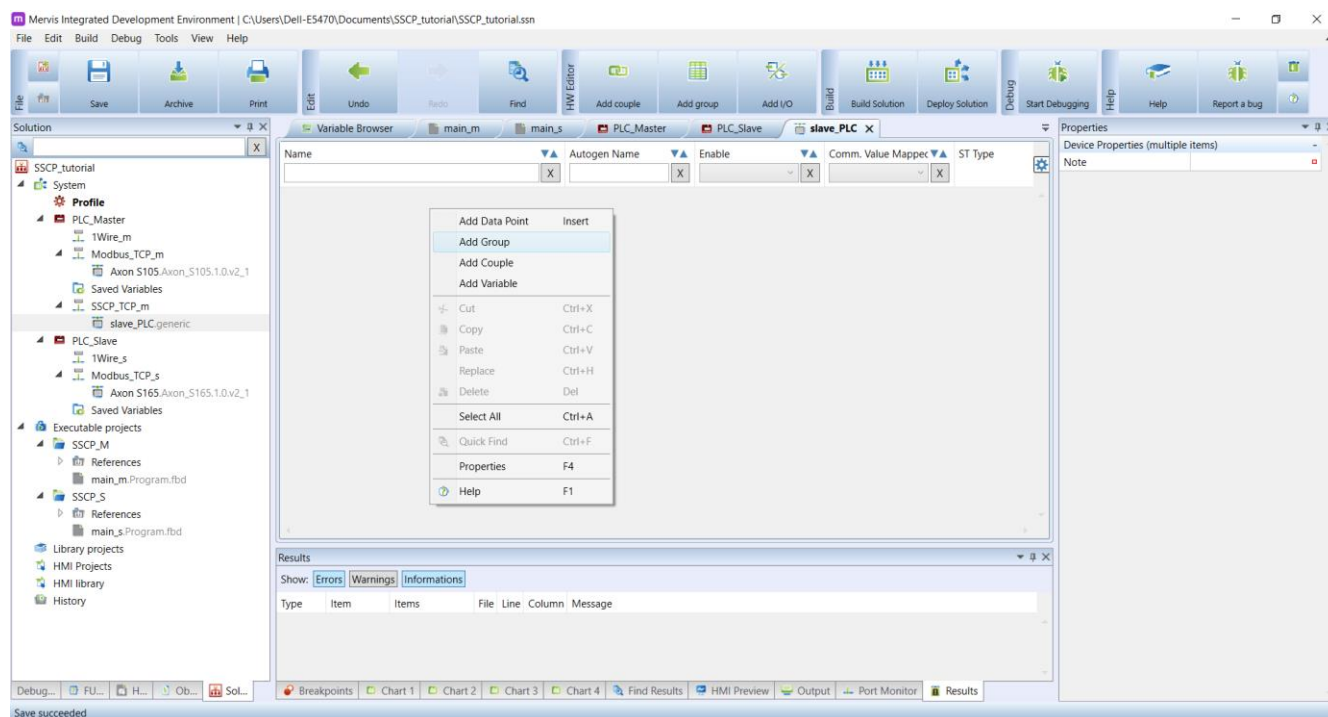


Zařízení pojmenujeme **slave_PLC** a pokud máme zařízení označeno, v pravém sloupci nalezneme **vlastnosti**, ty vyplňte podle následujícího obrázku. Všechny údaje naleznete ve **vlastnostech PLC_Slave**.

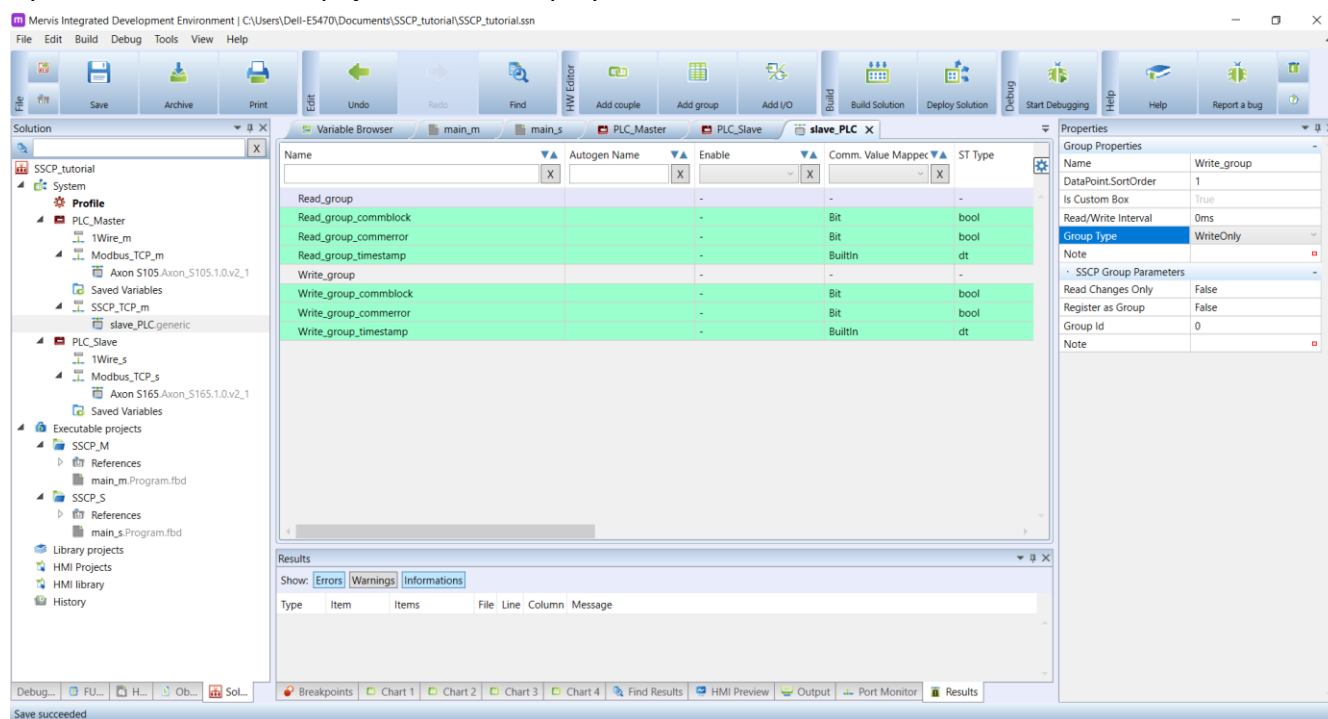


2 Proměnné, mapování a provázání

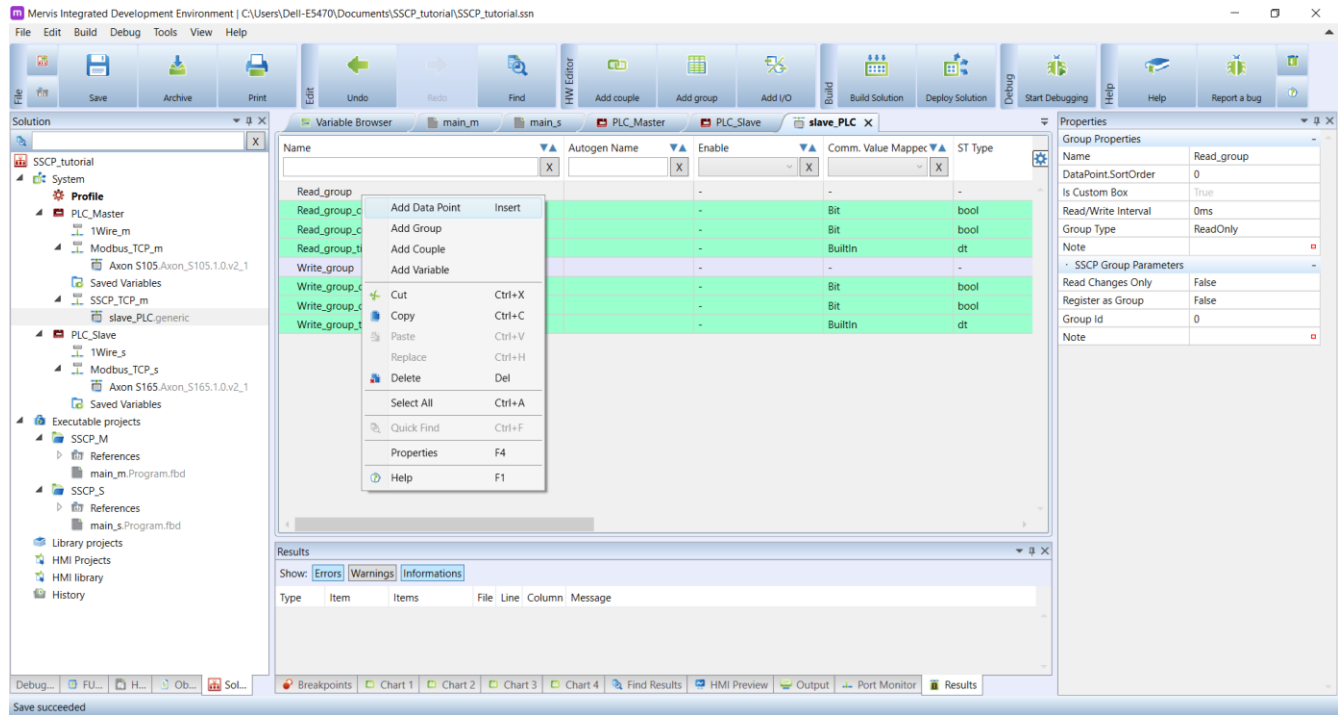
Ve vytvořeném zařízení vytvořte dvě **skupiny**, jednu na **čtení** a druhou pro **zápis**. Obdobně si je i pojmenujeme.



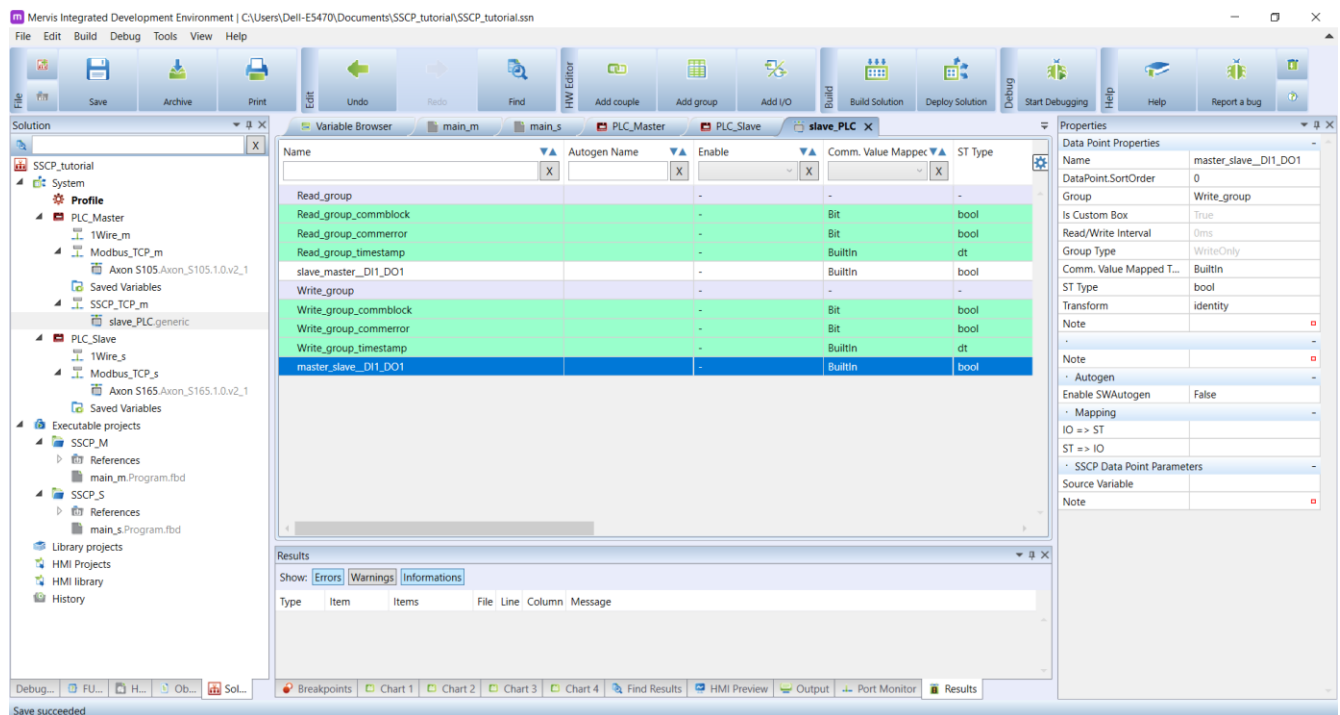
Vytvořené, nastavené a pojmenované skupiny vidíte na obrázku:



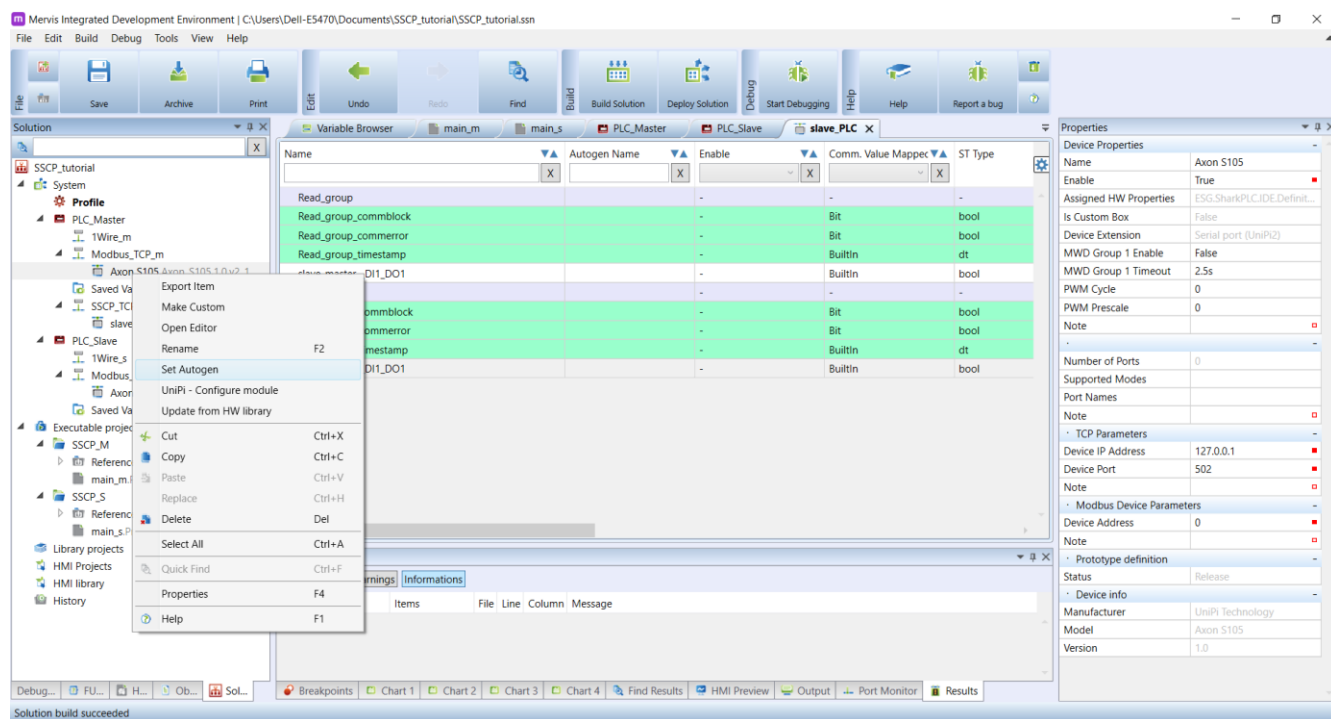
Pro každou skupinu přidejte pravým kliknutím jeden **datový bod**.



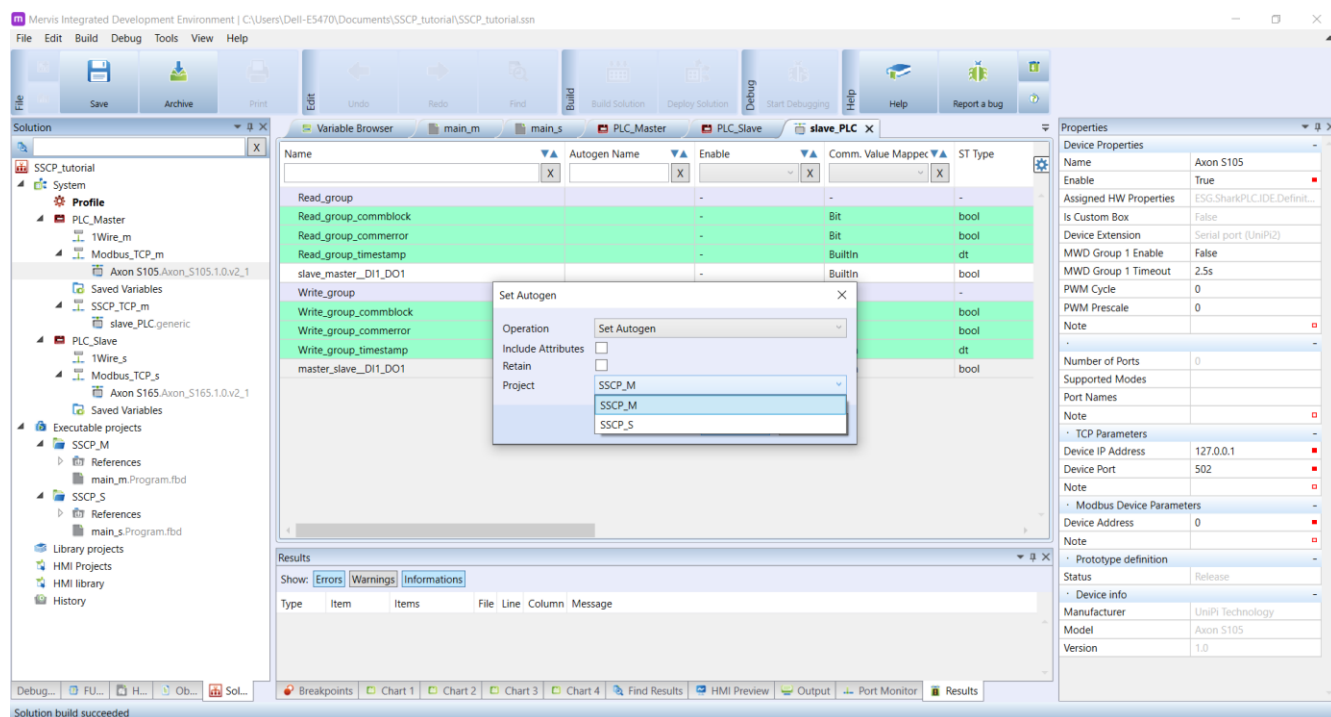
Po označení **datového bodu** naleznete v pravém sloupci jeho **vlastnosti**. Nastavte a pojmenujte dle obrázku:



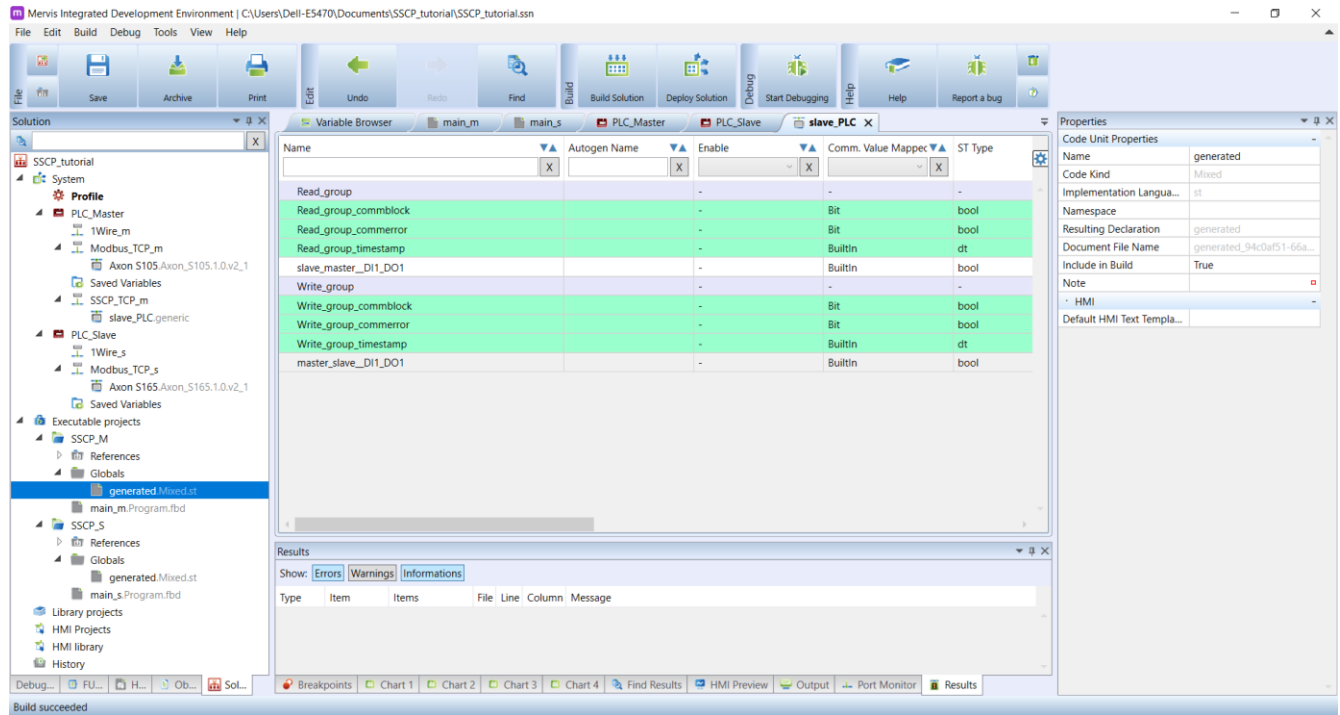
V tomto bodu použijeme **Autogen**. Pravým tlačítkem klikněte na každé **zařízení** pro každý **kanál** a zvolte **nastavit Autogen**.



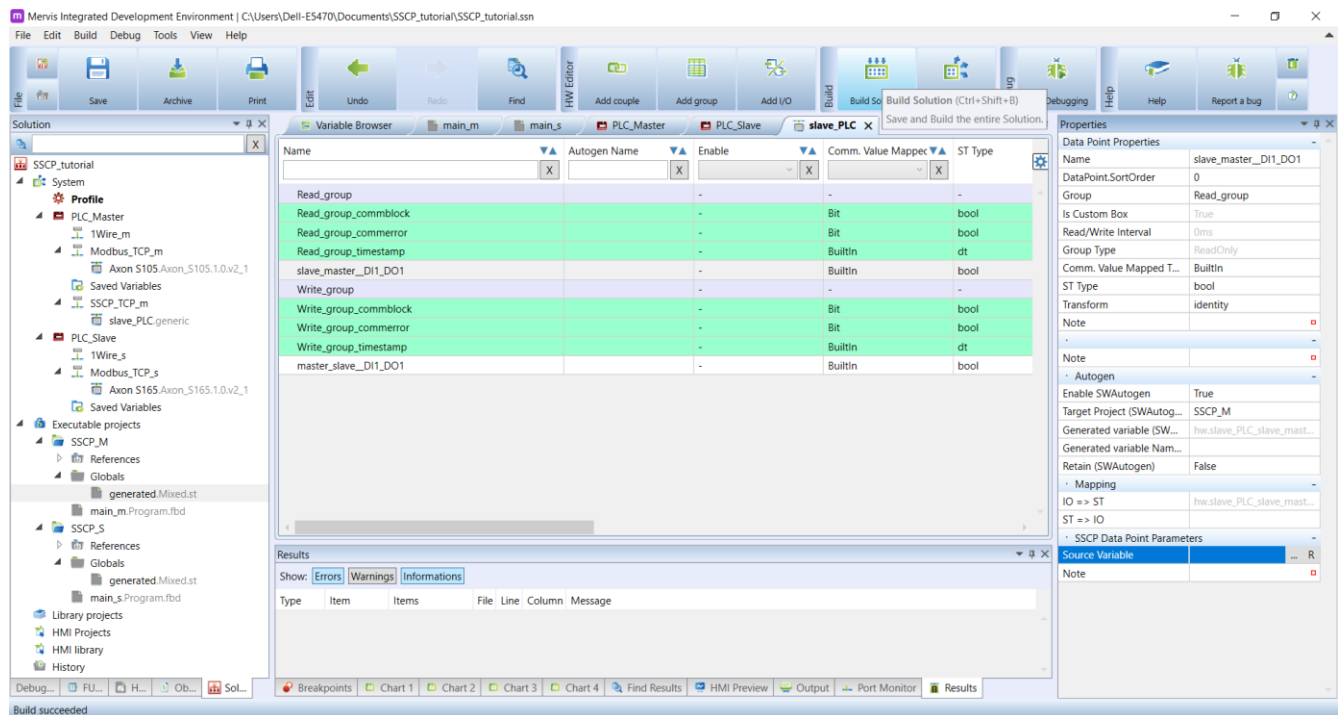
Hlavně dbejte na to, abyste všechny **zařízení** u **PLC_Master** přiřadili do **projektu SSCP_M** a vše u **PLC_Slave** do **projektu SSCP_S**.



Autogen vytvoří dva soubory **generated.Mixed.st** každý pro jeden **projekt** (pro jedno PLC).

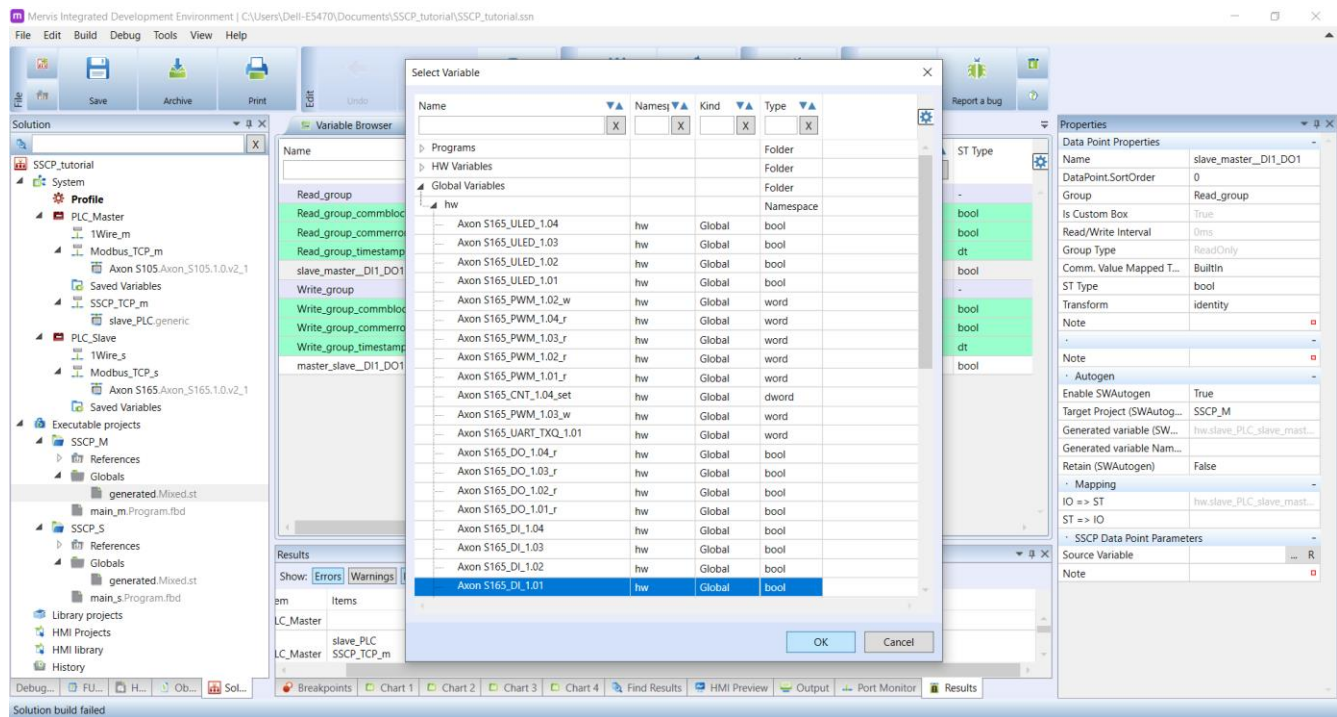


Ted' potrebujeme zadat **zdrojové proměnné** pro každý **datový bod**. Přejdeme do zařízení **slave_PLCL** a označíme **datový bod slave_master_DI1_DO1**. V pravém sloupci máme zobrazeny **vlastnosti** a předposlední položka je **zdrojová proměnná**. Abychom mohli proměnnou nastavit musíme ale nejprve **zkompilovat sestavu**. Znovu počítejte s vyspáním chybové hlášky, jelikož projekt stále není kompletní.

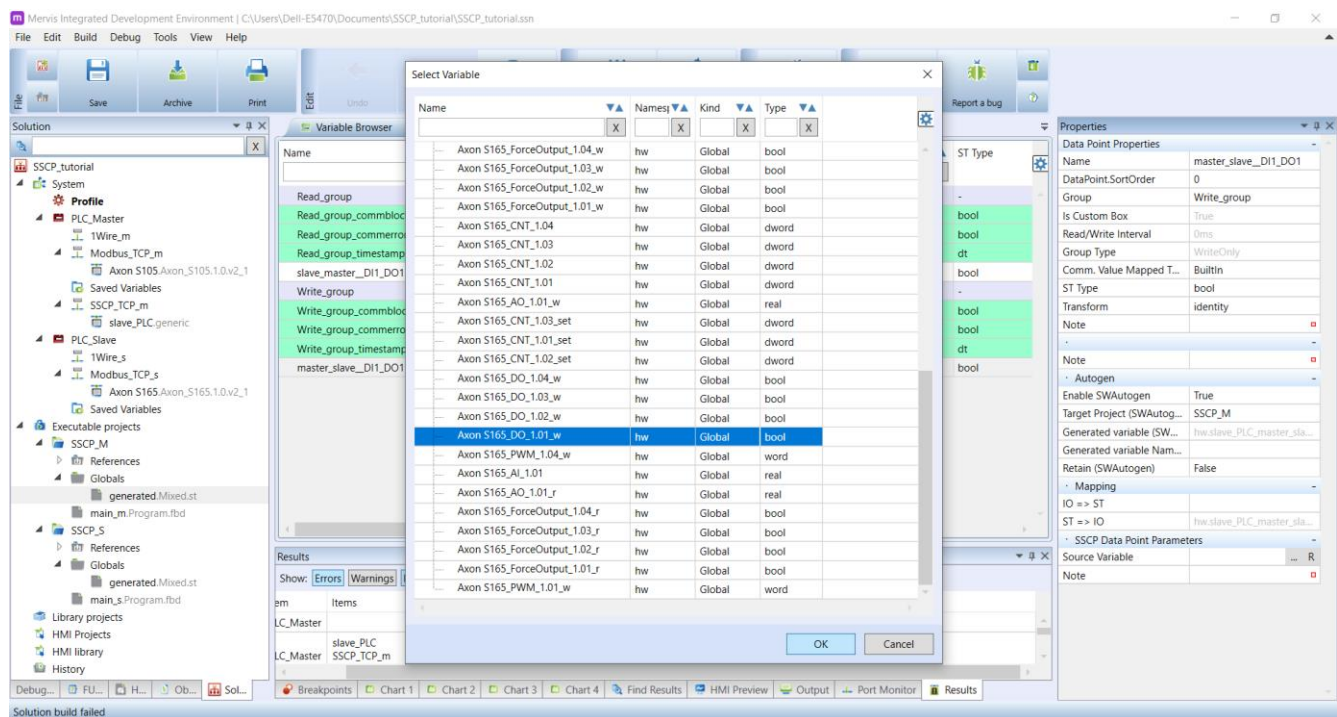


Po pokusu o zkompilevání se nám ale načtou **Autogenem** vytvořené proměnné a můžeme proto namapovat **zdrojovou proměnnou**.

Pro **slave_master_DI1_DO1** hledáme proměnnou **DI_1.01** z **PLC_Slave**.



Stejný postup aplikujte i na **datový bod master_slave_DI1_DO1**. Jen zde zadáte jako **zdrojovou proměnnou DO_1.01_w**



Nyní již máme vše ze strany základního nastavení hotové.

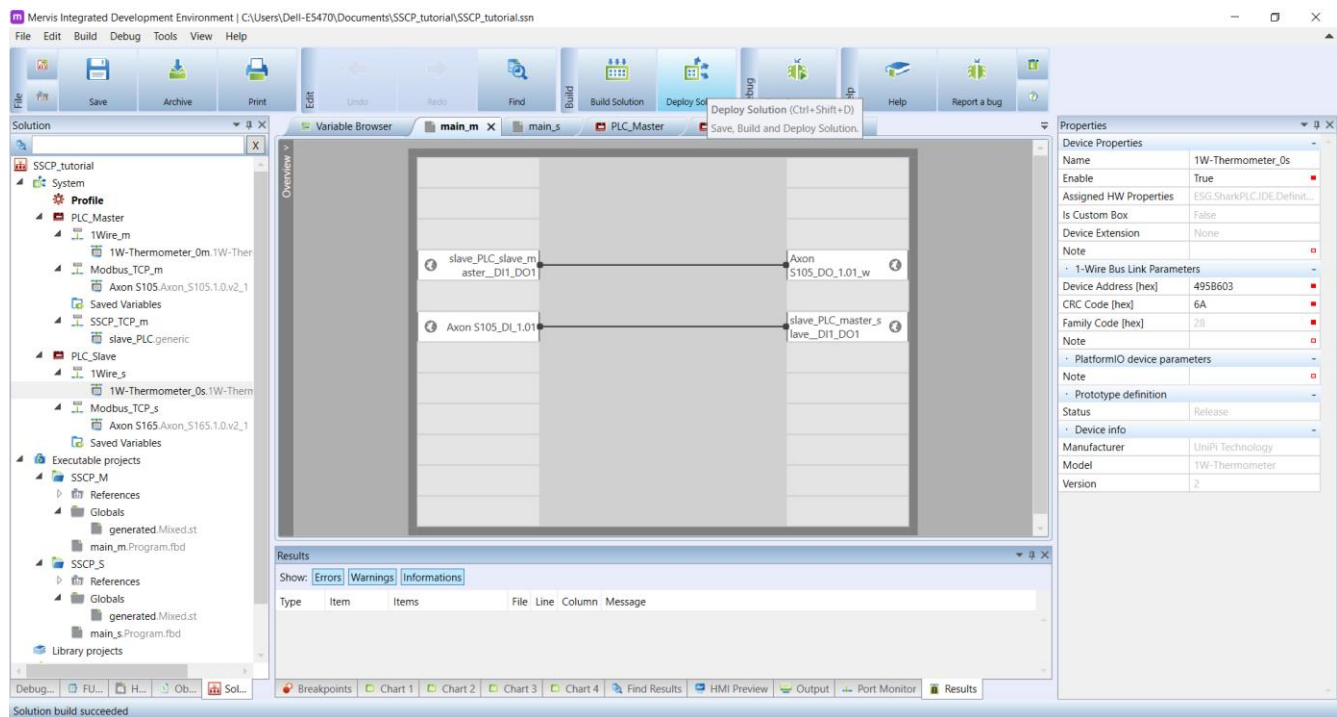
3 Nastavení programu (**main_m**)

Námi vytvořené proměnné vložíme do programu. Levá strana je pro vyčítání hodnot (slave -> master). Pravá strana slouží pro zápis hodnot (master -> slave). Pokud proměnné v seznamu nevidíte, znovu **zkompilujte sestavu** a proměnné se propíší.

Popis:

1. Přidáme proměnnou **slave_master_DI1_DO1** na vstup (vlevo) a proměnnou **DO_1.01_w** (**PLC_Master**) na výstup (vpravo) a propojíme
2. Přidáme proměnnou **master_slave_DI1_DO1** na výstup (vpravo) a proměnnou **DI_1.01** (**PLC_Master**) na vstup (vlevo) a propojíme

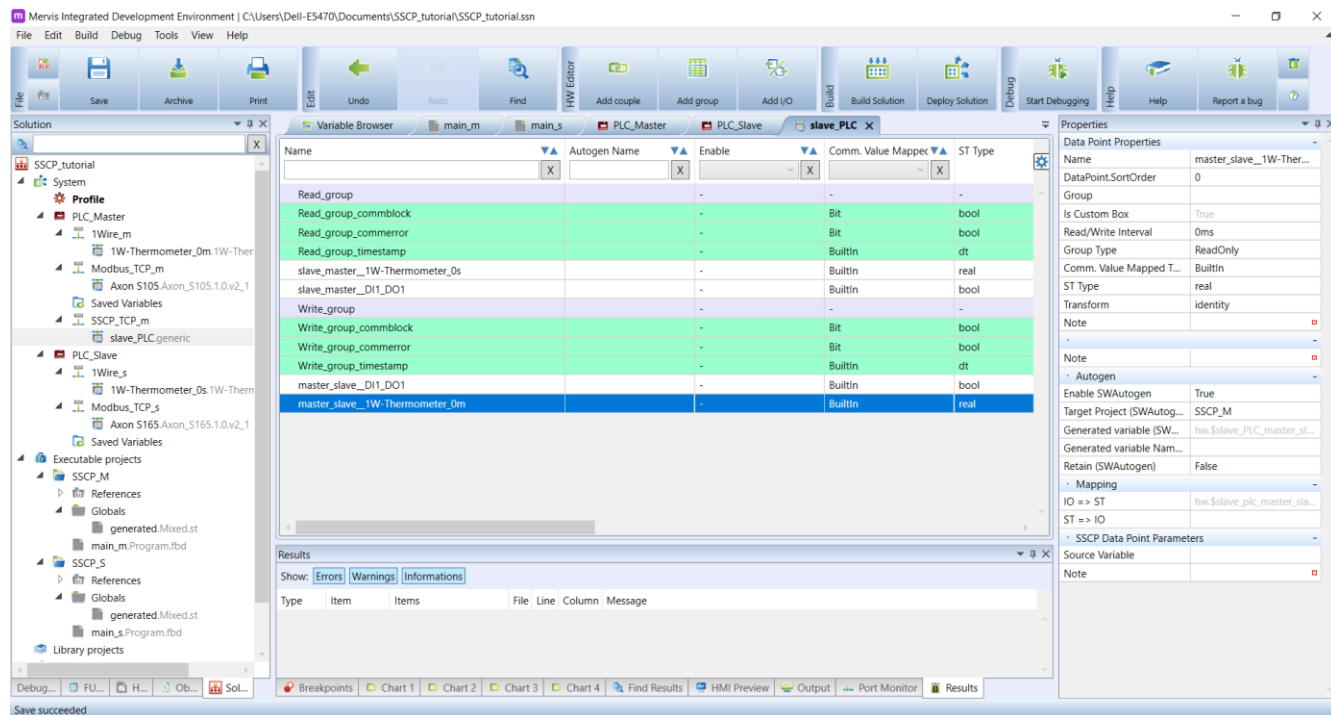
Klikněte na **nahrát sestavu**.



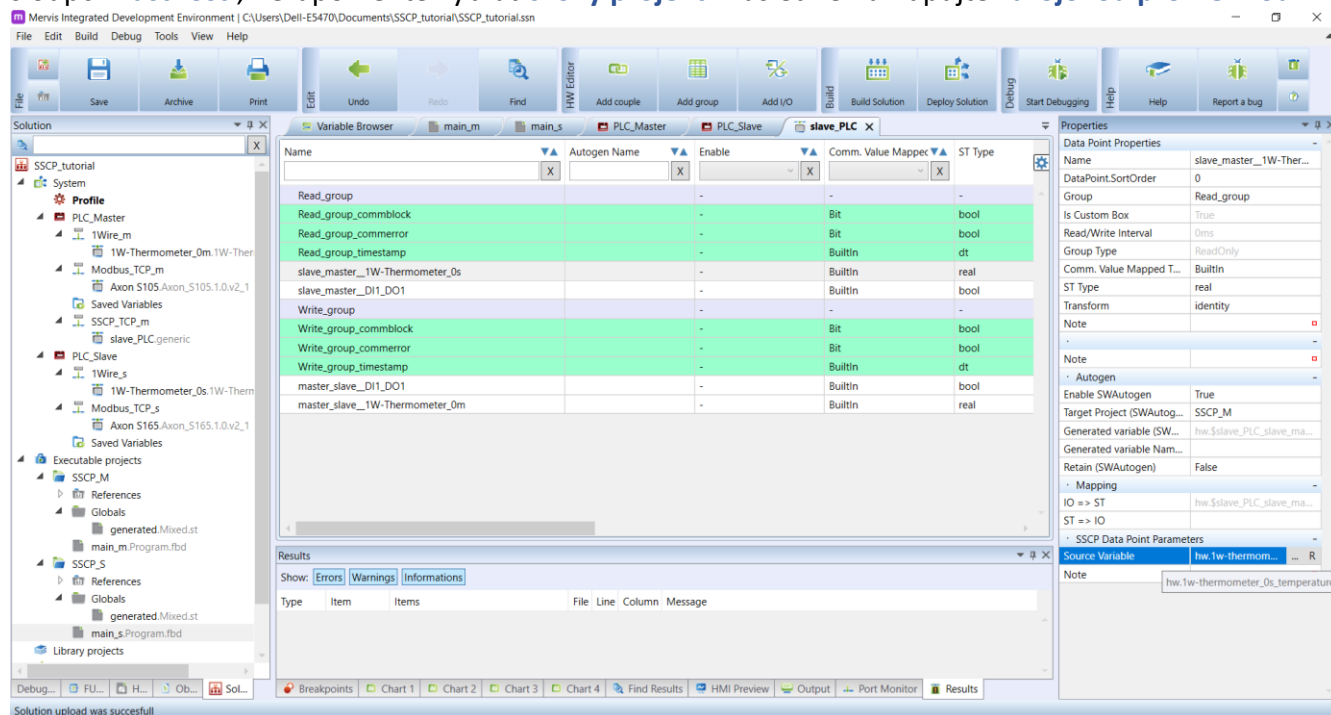
4 Pro pokročilé

Provedeme **detekci čidel 1-Wire** a **nastavíme autogen**. Podrobný návod naleznete v tutoriálu: [Attaching 1-Wire devices](#)

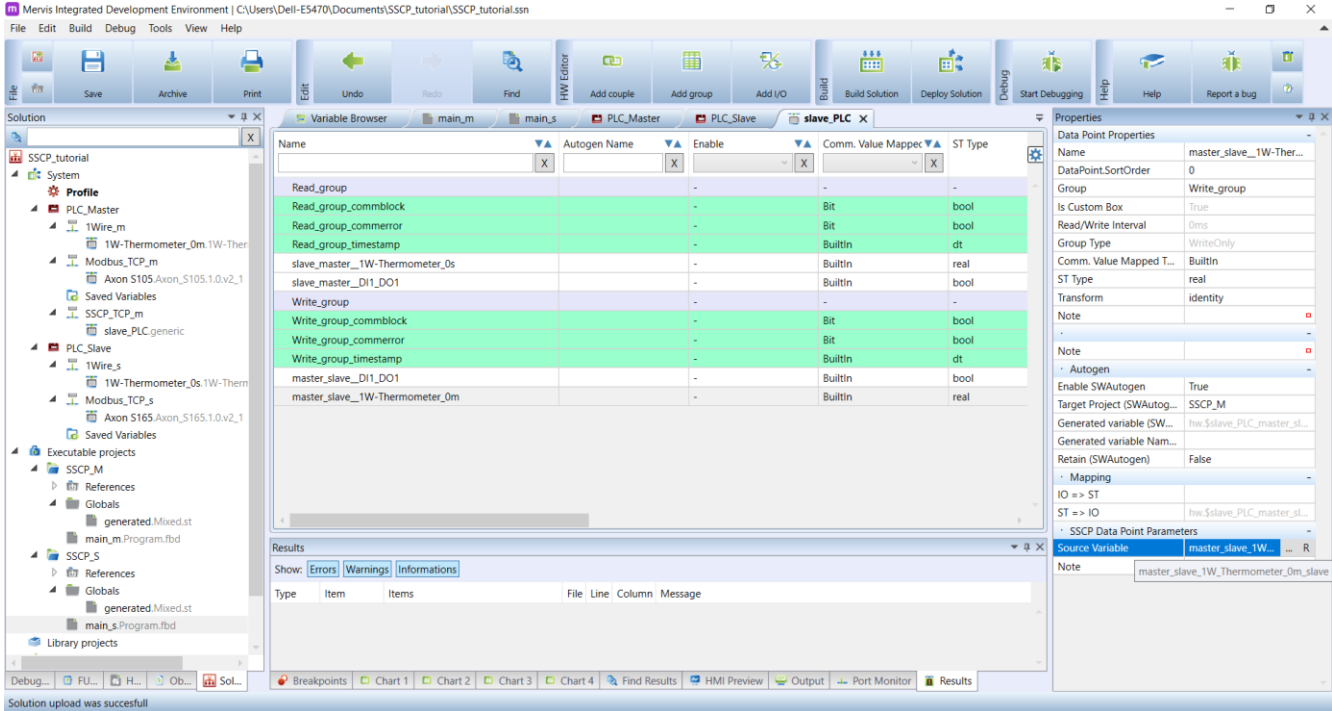
Vytvoříme další dva **datové body** a nazveme je podle obrázku:



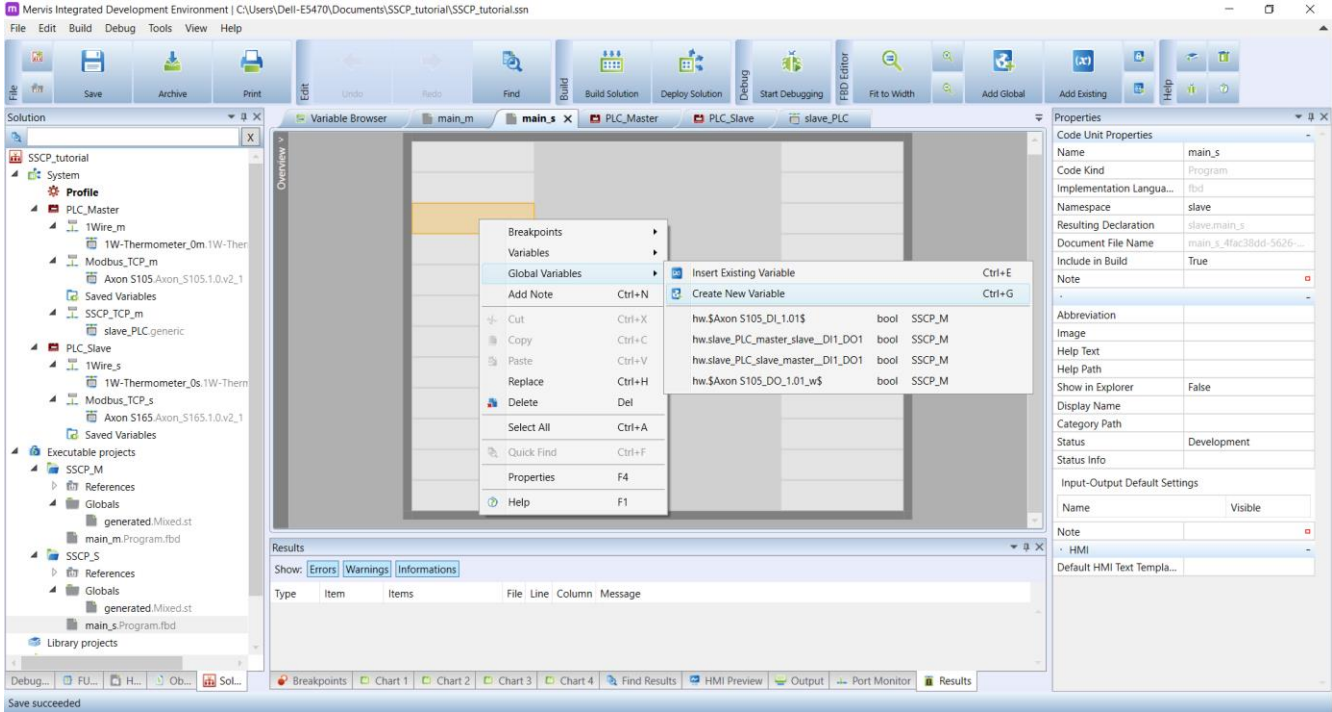
Oběma nastavíme **datový typ** a **skupinu**. **Autogen** pro tyto dvě proměnné zapněte manuálně ve sloupci **vlastnosti**, nezapomeňte vybrat **cílový projekt**. Následně namapujte **zdrojovou proměnnou**.



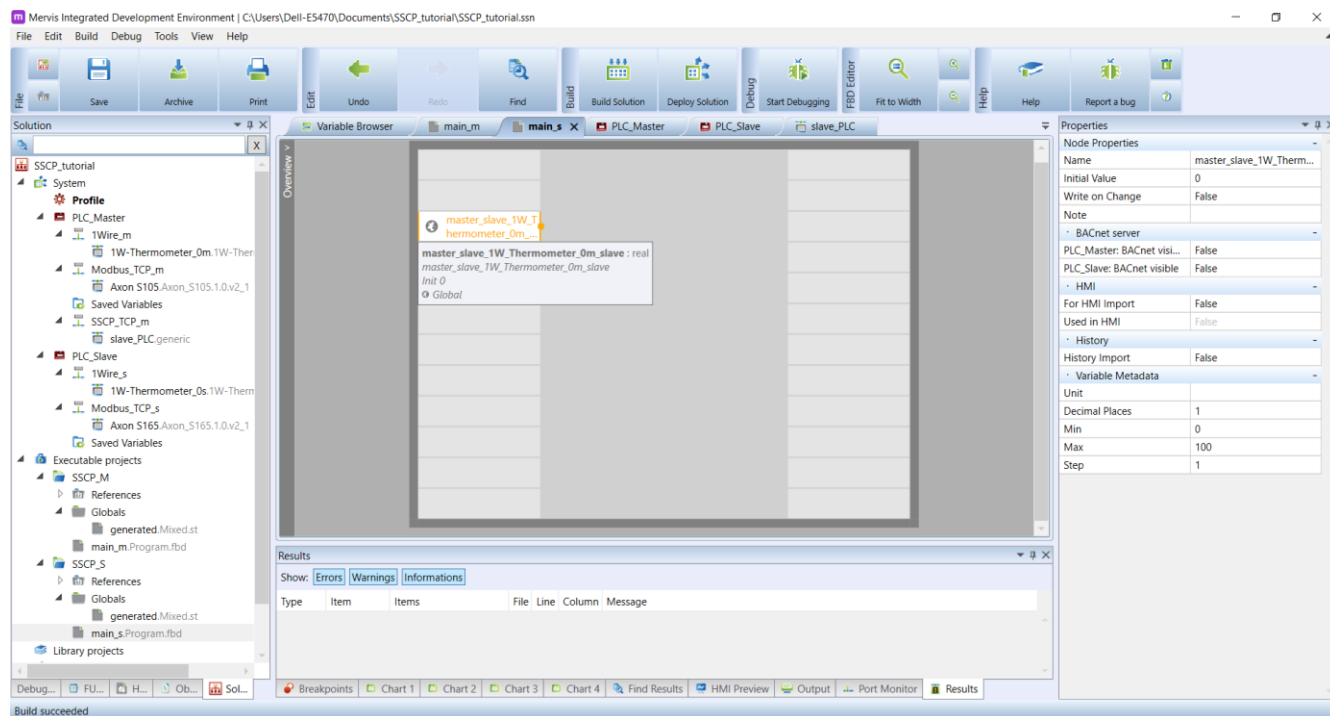
Opakujte i pro druhou proměnnou:



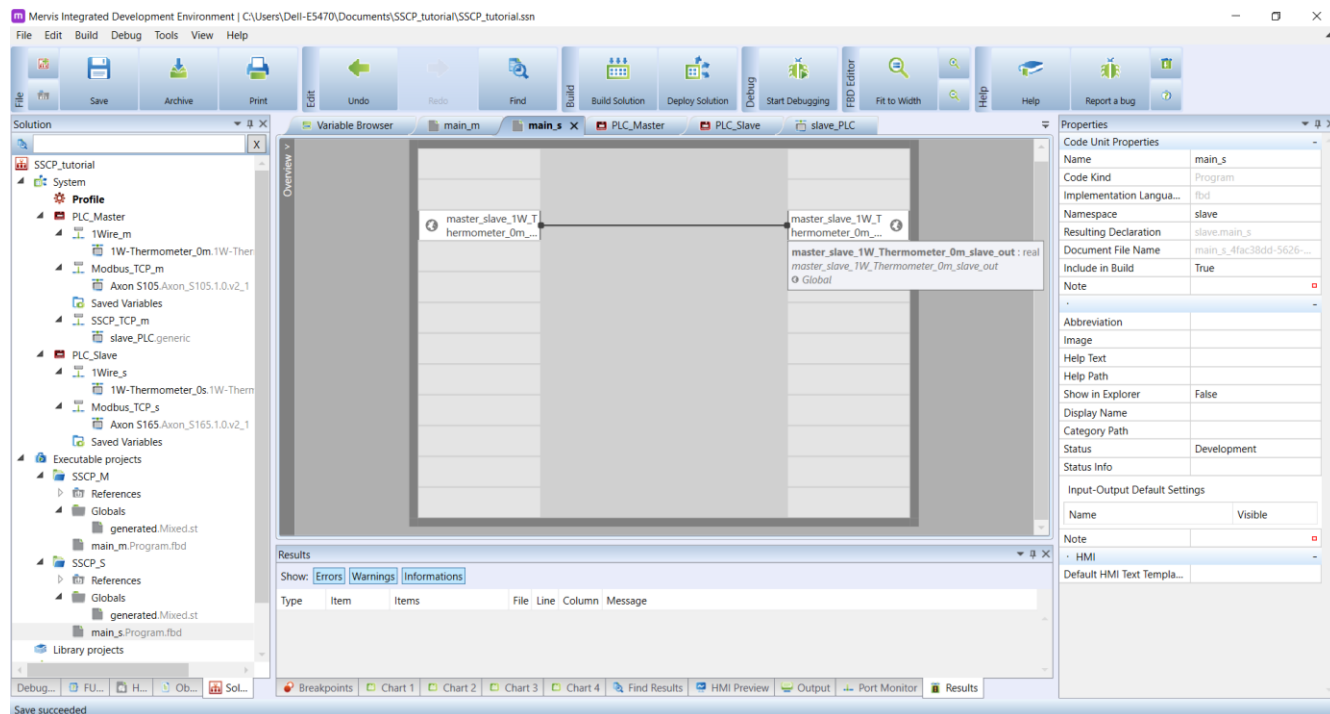
Přesuneme se do **main_s**. Otevřete dvojklikem na **main_s.Program.fbd** a vytvořte dvě globální proměnné (vstupní – vlevo, výstupní – vpravo).



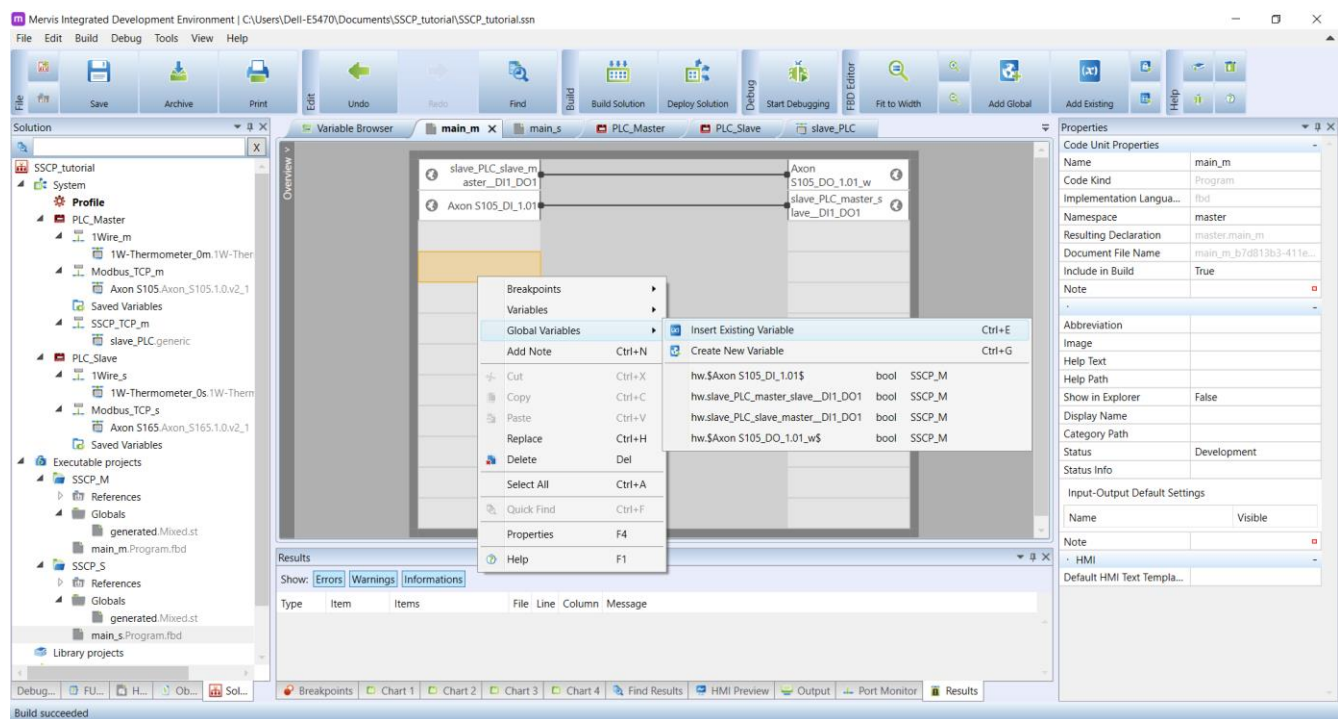
Vstupní proměnou pojmenujte následovně:



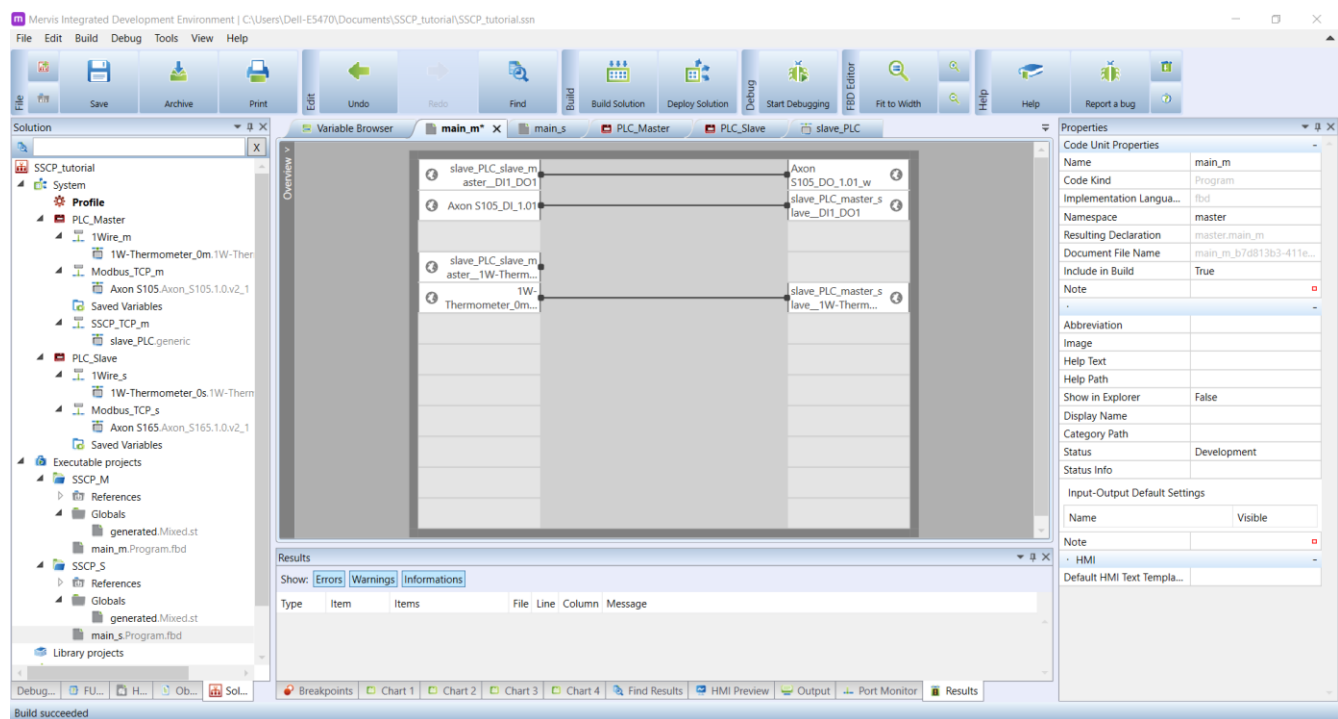
Výstupní proměnná je potřeba pouze pokud v **main_s** nemáte logiku, která by vstupní proměnou zpracovávala. **Mervis** totiž požaduje, aby vstupní proměnná byla „využita“ jinak ji nezahrne do **sestavy**. Bloky globálních proměnných propojíme. Znovu vhodně pojmenujte:



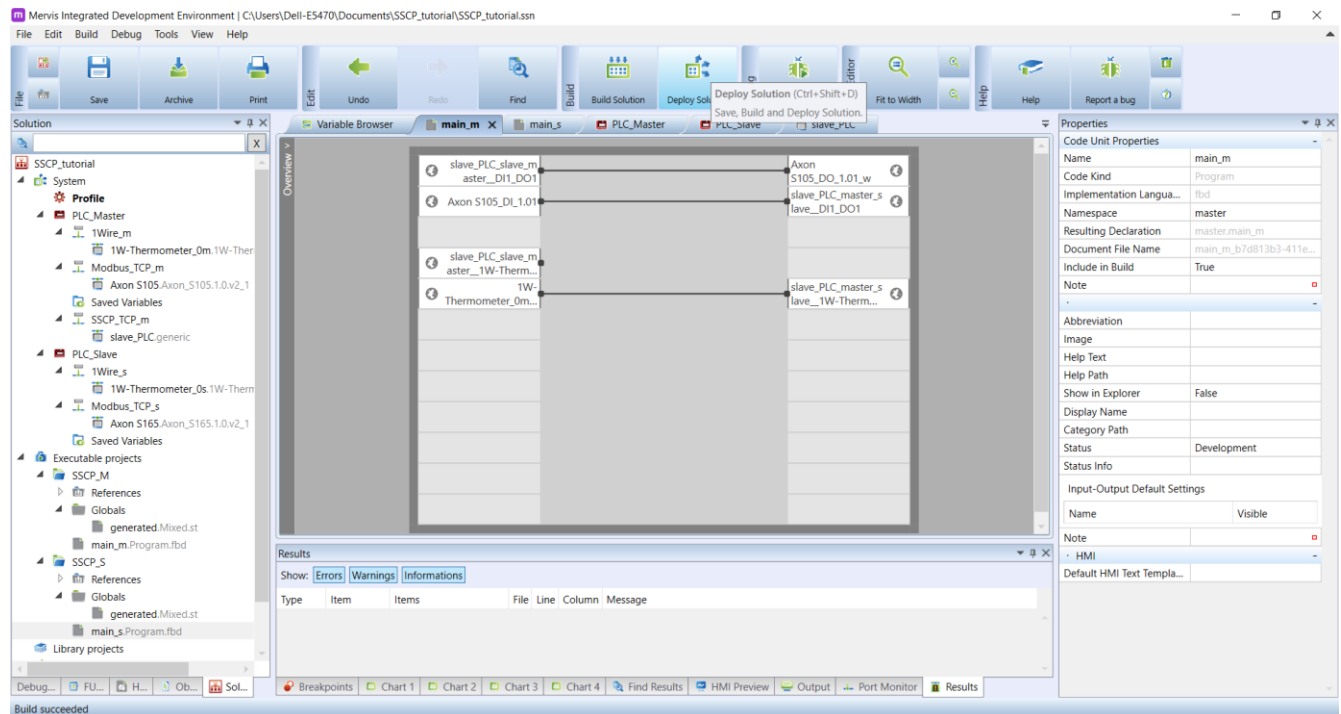
Přesuneme se do **main_m**. Otevřete dvojklikem na **main_m.Program.fbd** a vložte proměnné vytvořené v **slave_PLC**. (čtecí vlevo, zapisovací vpravo)



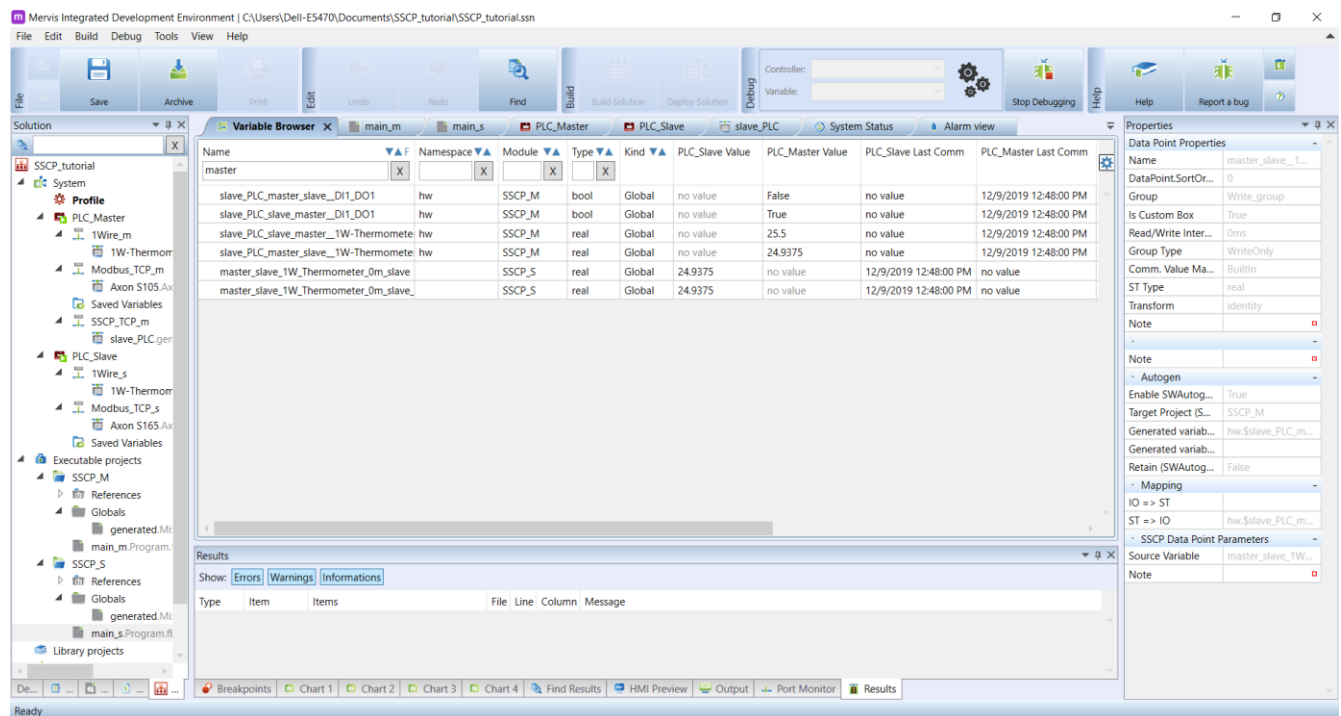
Vložte na vstupní sloupec ještě proměnnou s výstupem z 1-Wire čidla (**PLC_Master**) a propojte s výstupní proměnnou podle obrázku:



Všechny úpravy máme hotové. Klikneme proto na **nahrát sestavu**



V módu **ladění** (kliknout na **start ladění**) přejdeme na **prohlížeč proměnných** a využijeme filtr s textem **master**, nebo **slave**. Tím si vyfiltrujeme jen proměnné, které nás zajímají:



5 Výsledek

- Nyní je již vše nastaveno a obě PLC na vstup **DI_1.01** reagují tak, že propisují hodnotu na výstup **DO_1.01** druhého PLC.
 - Pro pokročilé: A vzájemně propisují si i teploty na čidlech (vše ale stále řídí **PLC_Master**)
 - Všechna logika a převážná část nastavení je provedena na **PLC_Master**.
 - Možné využití:
 - Vyčítání ze vstupů, nebo zápis na výstupy jiného PLC
- PRO POKROČILÉ:**
- Sdílení senzorů (jeden sensor pro více PLC)
 - Sdílení programový proměnných -> **PLC_Master** sleduje konkrétní proměnnou z **PLC_Slave (bool)** a ve chvíli, kdy **PLC_Slave** nastaví tuto proměnnou na **TRUE**. **PLC_Master** na tento podnět aktualizuje např. **real** proměnnou v **PLC_Slave**. **PLC_Slave** zařízení si proměnnou dále zpracovává podle svého programu **main_s**.
 - **PLC_Master** může proměnnou posílat do **PLC_Slave** i každou vteřinu, ale může dojít k tomu, že se zahltí komunikace a vše bude pomalé. Tuto variantu řešení jsem použil pro tento tutoriál, protože pro účelnost tohoto tutoriálu je logika v **PLC_Slave** irelevantní.