

1 Postupy při návrhu distribuovaných řídicích systémů

Tato kapitola se věnuje obecnému popisu kroků nutných pro návrh distribuovaných řídicích systémů. Také lokální řízený systém může využívat návrhu distribuovaných systémů, např. pro zlepšení přehlednosti návrhu nebo pro jednodušší návrh algoritmu řízení. Pro distribuované systémy řízení jsou důležité informační vazby, které jsou zajištěny ve většině případů pomocí ILAN a LAN. Této problematice se věnuje jedna z předchozích kapitol.

Návrh distribuovaného systému řízení lze rozdělit do těchto kroků:

- specifikace zadání řešené úlohy,
- distribuce algoritmů řízení,
- výběr technických prostředků,
- výběr informačních vazeb,
- tvorba aplikací pro jednotlivé úrovně.

1.1 Specifikace zadání řešené úlohy

Před vlastním návrhem řídicího systému jak po stránce programové, tak po stránce technických prostředků je nutné provést analýzu zadání a rozdělit celý algoritmus řízení, vyplývající se zadání, na dílčí části. Již při tomto bodu řešení návrhu distribuovaných řídicích systémů je přínosem znalost funkčních možností a komunikačních možností řídicích systémů. Tímto základním rozdělením a analýzou úlohy (zadání) lze získat na přehlednosti návrhu celé aplikace. Je možné rozdělení složitých algoritmů řízení na dílčí jednoduché algoritmy umístěné na jednotlivých úrovních řízení. Ideálním výsledkem může být rozložení složitěho algoritmu řízení na triviální algoritmy řízení, které se aplikují do řídicích systémů.

Podstatnou částí použitých řídicích systémů jsou komunikační rozhraní a jejich možnost použití při „RealTime“ řízení. Algoritmy v těchto systémech musí být vytvořeny s ohledem na bezpečný provoz celého systému i v případě výpadku komunikačních vazeb, které se využívají jak pro konfiguraci řídicího systémů, zadávání žádaných hodnot dílčího podsystému, tak pro monitorování provozních veličin.

Pro tuto realizaci návrhu existují nástroje i pro SCADA/MMI systémy umožňující distribuovaný návrh algoritmů podporující komunikaci řídicích systémů v síti. Patří zde např. v systému Control Web možnost tvorby local, remote a attach modulů. Tyto moduly aplikace mohou pracovat v síti a mají rozdílné vlastnosti z pohledu běhu celé aplikace. Podrobnější informace nalezneme v literatuře [Bílý 1999].

1.2 Distribuce algoritmů řízení

Podle základní analýzy zadání provádíme distribuci jednotlivých řídicích algoritmů do řídicích systémů s ohledem na optimalizaci chodu celého distribuovaného systému.

Příkladem může být distribuce algoritmu pro polohování krokového motorů, který pomocí mechanického převodu polohuje pracovní nástroj (nůž). Tento algoritmus může být implementován do řídicího systému na nejnižší úrovni řízení, který dostává příkazy (pro dosažení koncové pozice nástroje) z nadřazené úrovně řízení (koordinace pohybu nástroje v osách x , y a z je řešena jedním algoritmem např. v jednočipovém počítači). Na vyšší úrovni řízení např. na IPC je realizována úloha pro monitorování a konfiguraci.

Druhým způsobem distribuce řízení může být koordinace algoritmů na vyšší úrovni řízení a jednočipové počítače na nejnižší úrovni řízení vykonávají dílčí příkazy algoritmů koordinace pohybu v osách x , y a z *vyšší úrovně řízení*.

1.3 Výběr technických prostředků

Základním kritériem pro výběr technických prostředků při návrhu distribuovaných systémů řízení je specifikace jejich vstupů a výstupů, komunikačních rozhraní, podpora meziobvodové komunikace a v neposlední řadě rychlost vykonávání navržených algoritmů a podpora programování ve vyšších programovacích jazycích např. programovací jazyk C. Při výběru technických prostředků může být vodítkem také podpora v podobě knihoven pro jednotlivé periferie (A/D, PWM, I2C, SPI, ...) nebo podpora komunikačních rozhraní (CAN) pro vyšší programovací jazyky.

Následně je nutné výkonově přizpůsobit a galvanicky oddělit I/O od řízené technologie. Komerční řídicí systémy, jako jsou např. PLC, toto oddělení a částečné přizpůsobení již obsahují.

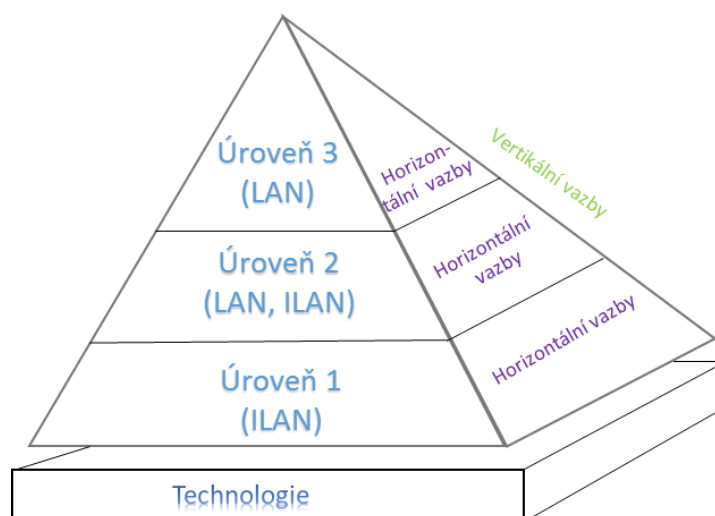
Standardním vybavením řídicích systémů na nejnižší úrovni řízení jako jsou PLC, jednočipové počítače je:

- Konfigurovatelné **diskrétní vstupy/výstupy**. Počet je většinou násobkem 8 bitů (1 byte) tj. 8, 16, 24, Směr (zda se jedná o vstup nebo výstup) je většinou konfigurovatelný z programového prostředí.
- **Analogové vstupy**. Systémy většinou obsahují jeden A/D převodník, kterému předchází multiplex kanálů. V závislosti na počtu čtených vstupů se mění vzorkovací frekvence. Se zvyšujícím se počtem čtených kanálů se snižuje vzorkovací frekvence. Další důležitou vlastností analogových vstupů je zda se jedná o diferenční nebo single-ended vstup. S tím souvisí jejich počet (single-ended=2*diferenciál).

- **Porovnávací vstupy.** Některé řídicí systémy mají zabudovány ve vstupních obvodech komparátory – lze tedy porovnávat analogový signál s předem nastavenou úrovní.
- **I2C, SPI** jsou komunikační rozhraní určená pro meziobvodovou komunikaci např. s pamětí, A/D převodníkem, snímači,
- **PWM** výstup je většinou náhradou za analogový výstup, pokud není součástí systému, a slouží pro ovládání akčních členů. Možností je také připojení D/A převodníku k některé z diskretních bran řídicího modulu nebo přes některé rozhraní I2C, SPI,
- **UART/USART** je komunikační rozhraní, které lze využít ve většině případů k připojení řídicího systému k nadřazené úrovni. Toto rozhraní může být také použito pro čtení dat ze snímačů nebo zápis hodnoty akčního zásahu pro akční člen apod. Podmínkou je podpora tohoto rozhraní a použitého protokolu na obou stranách komunikace.
- **Podpora průmyslové sběrnice.** V řídicím systému může být implementované rozhraní některé z průmyslových sběrnic např. CAN bus, které lze opět využít pro správu (monitorování) systému nebo pro sběr dat ze snímačů, koordinační vazby apod.

1.4 Výběr informačních vazeb

Informační vazby mezi jednotlivými řídicími podsystémy jsou pro distribuované systémy zásadním nástrojem pro přenos technologických dat a zadávání parametrů pro dílčí algoritmy. Z pohledu distribuovaných systémů řízení a hierarchické struktury řízení jsou dva základní typy vazeb:



Obr. 1-1 Vazby hierarchické struktury řízení

Horizontální vazby – jsou realizovány s využitím ILAN a LAN. Výběr typů sítí je většinou podmíněn úrovní nasazení, tj. zda se na dané úrovni mohou objevit rušivé signály (technologická úroveň), pak se využívají ILAN a v případě, že se dostáváme v návrhu distribuovaných systémů řízení na vyšší úrovně, vzdálenější od technologie, můžeme využívat v návrhu třeba LAN. Tyto vazby přenáší informace mezi jednotlivými řídicími systémy na stejné úrovni a koordinují algoritmy v řídicích systémech na stejné úrovni řízení.

Vertikální vazby – pro využívání ILAN a LAN platí stejné pravidlo jako u horizontálních vazeb pro rušení přenosu. Tyto vazby přenáší informace mezi řídicími systémy na rozdílných úrovních řízení hierarchické struktury řízení. Informační hodnota na těchto vazbách se většinou snižuje směrem k vyšším úrovním řízení (např. pro třetí úroveň řízení je důležitá jen aktuální rychlost výrobní linky, ale ne parametry jednotlivých dílčích regulátorů).

1.5 Tvorba aplikací pro jednotlivé úrovně řízení

V závislosti na úrovni řízení a použitých technických prostředcích se využívají různá prostředí pro programování dílčích algoritmů pro jednotlivá zařízení. Pro programování aplikací na nejnižší úrovni řízení se velmi často využívá jazyk C, protože má po assembleru nejbližší k hardwarové konfiguraci řídicího systému (brány I/O, registry, ...). Existují různé vývojové prostředí pro programování řídicích systémů podporující programování v jazyce C. Další možností je například programování v jazyku Basic. Příkladem může být vývojové prostředí pro programování jednočipových počítačů firmy MikroElektronika MikroC nebo MikroBasic. Dalším příkladem je prostředí firmy Microchip MPLAB podporující programování jak ve strojovém kódu, tak programování ve vyšších programovacích jazycích C nebo Basic. Na nejnižší úrovni řízení mohou být i speciální prostředí například pro programování PLC. Zde jsou z důvodu kompatibility mezi jednotlivými PLC použity základní typy programovacích jazyků:

- IL instruction list,
- LD ladder diagram,
- FBD function block diagram,
- ST strukturovaný text (Pascal),
- SFC sekvenční programování.

Programování pomocí IL je v principu programování v assembleru tj. programování pomocí instrukcí procesoru PLC. Jedná se o základní instrukce přesunu, aritmetické instrukce (sčítání, odčítání, ...), skokové instrukce (podmíněný a nepodmíněný skok), logické instrukce (AND, OR, ...), Tento způsob programování je velmi náročný na programátora, protože musí dokonale ovládat celou sadu instrukcí daného procesoru. Programování algoritmu pomocí LD je podobné sestavování logické úlohy pomocí spínačů atd. Popis jednotlivých způsobů programování lze nalézt v odborné literatuře zabývající se touto problematikou.

