



AMK
Control your Motion.

PLControl

Лекции

Компютърни системи за управление (КСУ)

Летен семестър 2008/2009

Любомир Борисов

Управител

“ПЛ Контрол” ООД

Симеон Трифонов

Ръководител отдел “Развитие”

Мариян Няголов

Тестов инженер

“АМК – Задвижваща и управляваща техника” ЕООД



Катедра “Автоматика,
информационна и управляваща
техника”

АИУТ/КСУ Летен семестър 2008/2009



AMK

Control your Motion.

PLControl

Лектори:

- Любомир Борисов

- ПЛ Контрол ООД
- ул. "Македония" №33
- Телефон 066 873722, 807424
- Email: lyubomir.borisov@start.bg



- Симеон Трифонов

- АМК ЕООД
- ул. "Генерал Николов" №1
- Телефон 066 819125
- Email: simeon.trifonov@amk-drives.bg



Ръководител на лабораторните упражнения:

- Мариян Няголов

- АМК ЕООД
- ул. "Генерал Николов" №1
- Телефон 066 819 144
- Email: mariyan.nyagolov@amk-drives.bg



Катедра "Автоматика,
информационна и управляваща
техника"



AMK
Control your Motion.

PLControl

Разпис на занятията:

Седмица	Понеделник	Вторник	Сряда	Четвъртък	Петък
16.02 – 20.02	8-18	8-18	8-18	8-18	8-18
23.02 – 27.02	8-18	8-13	8-18		



Катедра “Автоматика,
информационна и управляваща
техника”

АИУТ/КСУ Летен семестър 2008/2009



AMK
Control your Motion.

PLControl

Изпит:

- *Писмен изпит*
- *Продължителност 2 часа*
- *05.Март.2009г зала 3315 9:00 часа*
- *Възможност за ползване на помощни материали (с изключение на комуникация от всякакъв род)*
- *Заклучително събеседване за прецизиране на оценката*



Катедра “Автоматика,
информационна и управляваща
техника”

АИУТ/КСУ Летен семестър 2008/2009



AMK
Control your Motion.

PLControl

Помощни средства:

- Копие на използваните слайдове в Интернет
- Техническа документация на продуктите на AMK
- Друга препоръчвана по време на лекциите литература



Катедра “Автоматика,
информационна и управляваща
техника”

АИУТ/КСУ Летен семестър 2008/2009



AMK

Control your Motion.

PLControl

Съдържание на учебния материал:

А. ЛЕКЦИИ

Модул I. Съставни елементи на електрозадвижващите системи:

- Електрически двигатели. Основни конструкции и принципи.
- Обратни връзки по позиция.
- Честотни инвертори и сервозадвижвания.
- Полево ориентиран контрол на променливотокови двигатели.
- Режимы на работа на сервозадвижване.

Модул II. Промислени интерфейси:

- Понятие за промишлен комуникационен интерфейс. Особенности.
- Point-to-point комуникация.
- Modbus.
- CANopen.
- ProfiBus, SERCOS.
- Real-time Ethernet базирани протоколи.

Модул III. Индустиални системи за управление:

- Основни понятия и принципи.
- Видове програмни единици.
- Езици за програмиране.
- Библиотеки с готови компоненти.
- Визуализации.



Катедра “Автоматика,
информационна и управляваща
техника”



AMK
Control your Motion.

PLControl

Съдържание на учебния материал:

В. ЛАБОРАТОРНИ УПРАЖНЕНИЯ

Към Модул I: Съставни елементи на електрозадвижващите системи:

- Базово параметризиране на сервозадвижване.
- Режимы на работа и настройка на сервозадвижване
- Настройка и параметризиране на регулатор на позиция.

Към Модул II: Промислени интерфейси:

- Конфигуриране на CANopen мрежа.
- Синхронно движение на цифрови серво-оси без механична връзка.
- Използване на промишлените интерфейси за зареждане на програмното осигуряване на индустриалните системи за управление.

Към Модул III: Индустриални системи за управление:

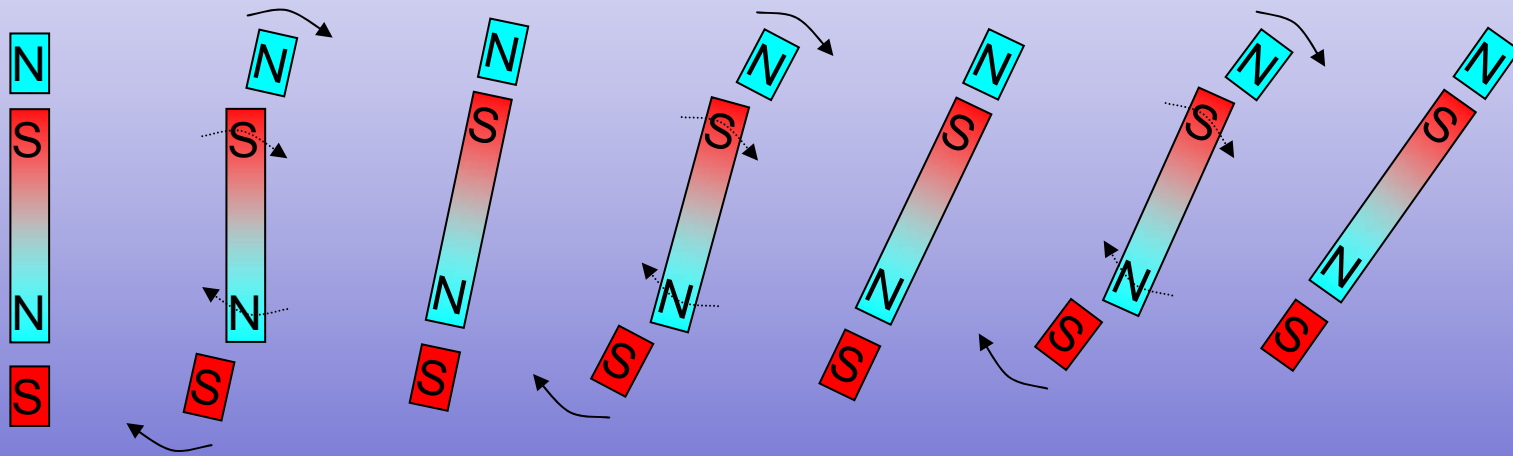
- Запознаване със средата за програмиране CoDeSys.
- Използване на вътрешни и външни библиотечни модули.
- Настройка, трасиране и отстраняване на грешки в потребителски програми.



Катедра “Автоматика,
информационна и управляваща
техника”

АИУТ/КСУ Летен семестър 2008/2009

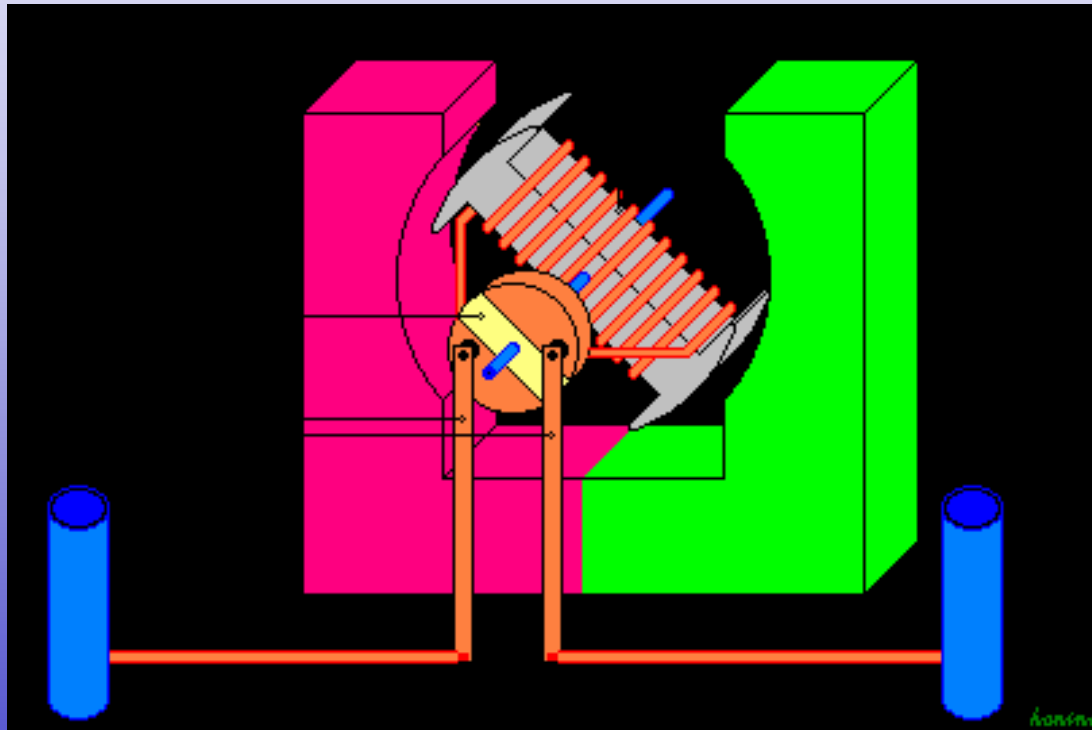
Принцип на работа на въртящата се електрическа машина:



Любомир Борисов

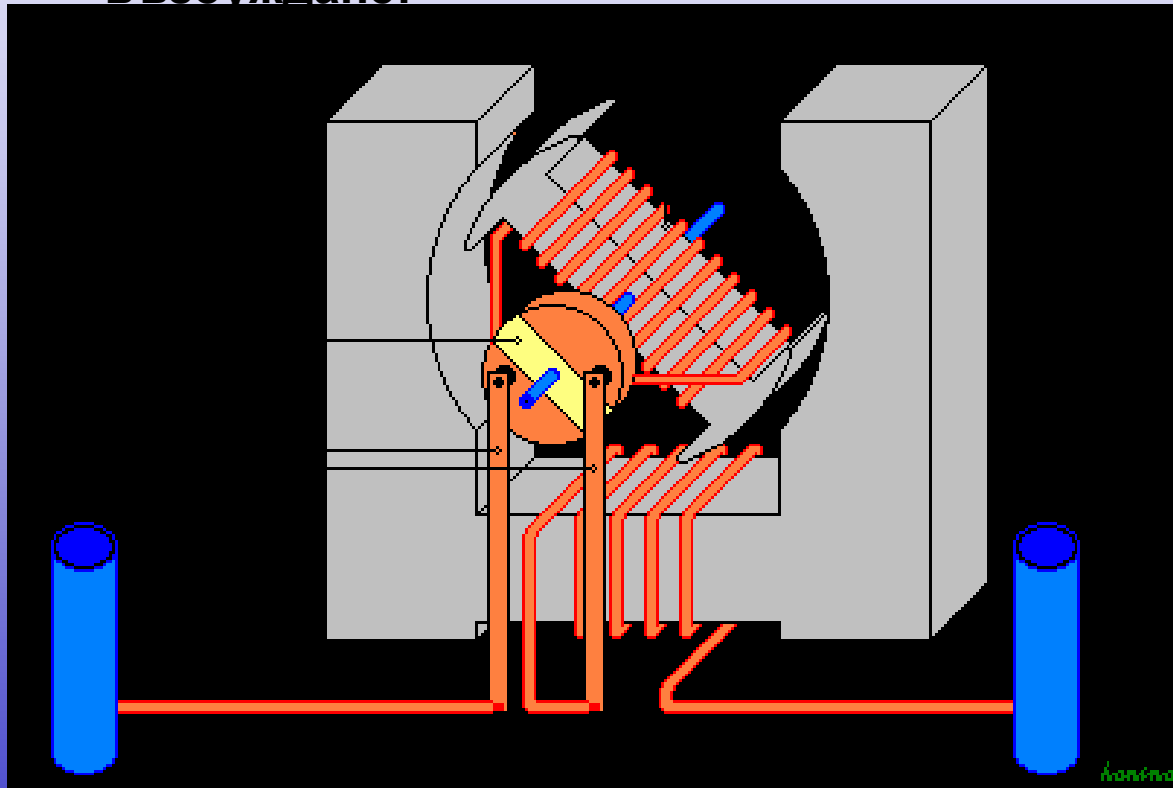


Постояннотоков електродвигател с постоянно възбудане:



Любомир Борисов

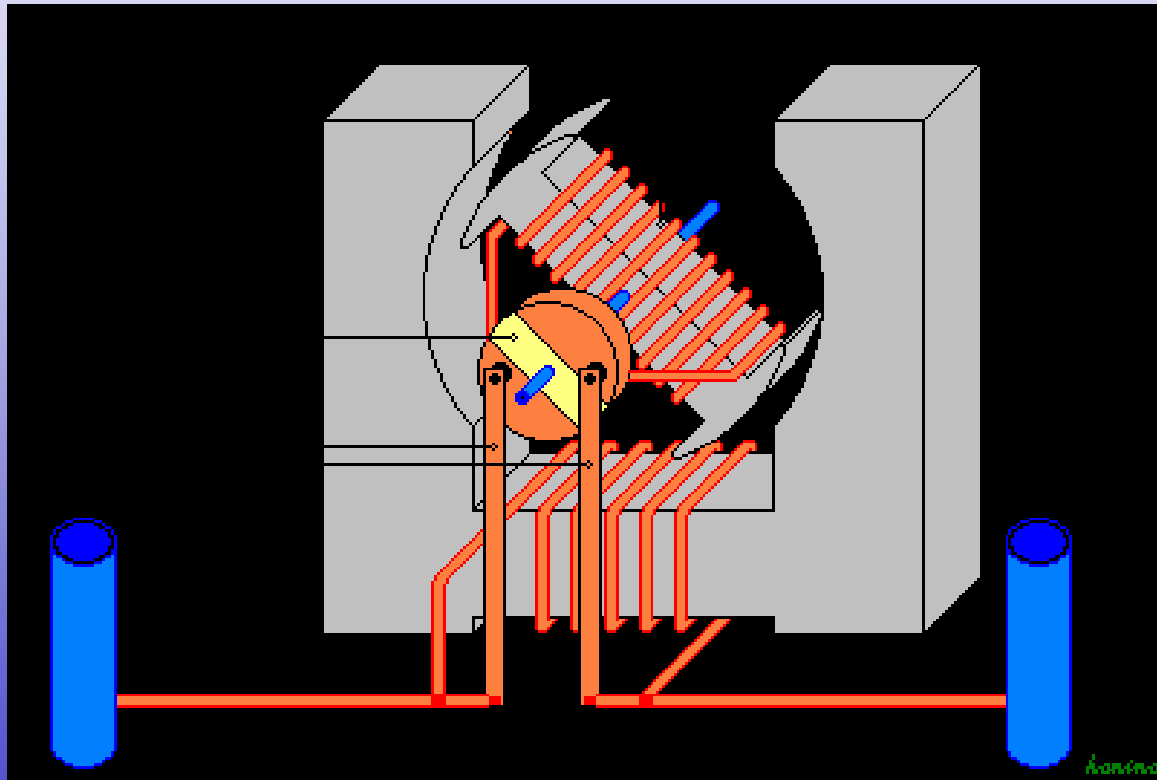
Постояннотоков електродвигател с последователно възбуждане:



Любомир Борисов



Постояннотоков електродвигател с паралелно възбуждане:

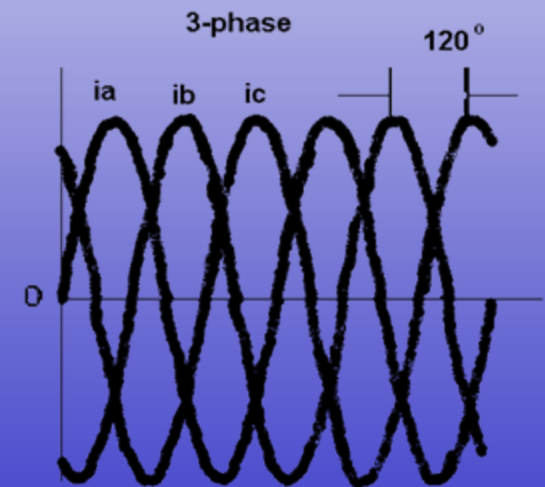
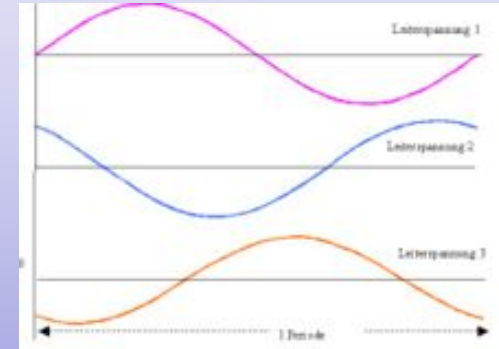
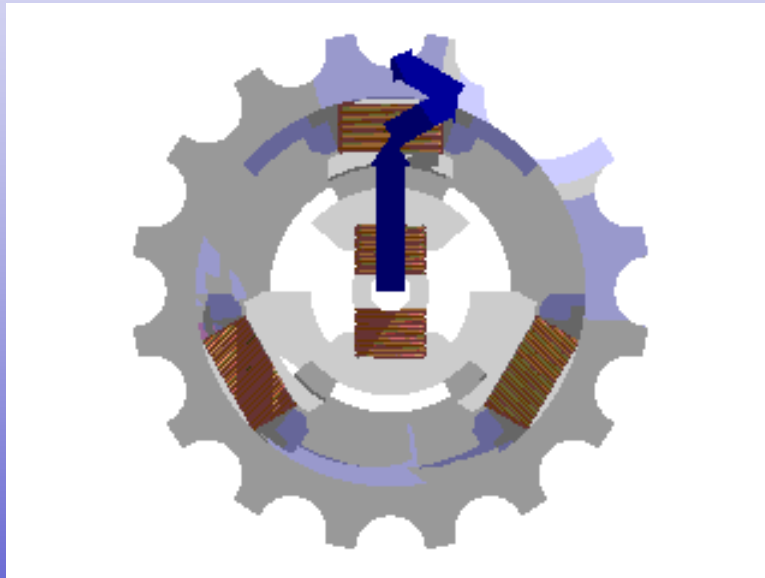


Любомир Борисов



Електрически двигатели. Основни конструкции и принципи.

Променливотокови електродвигатели:



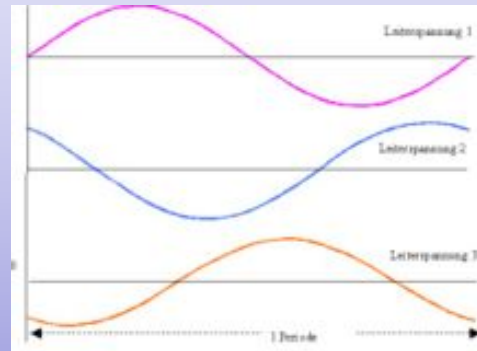
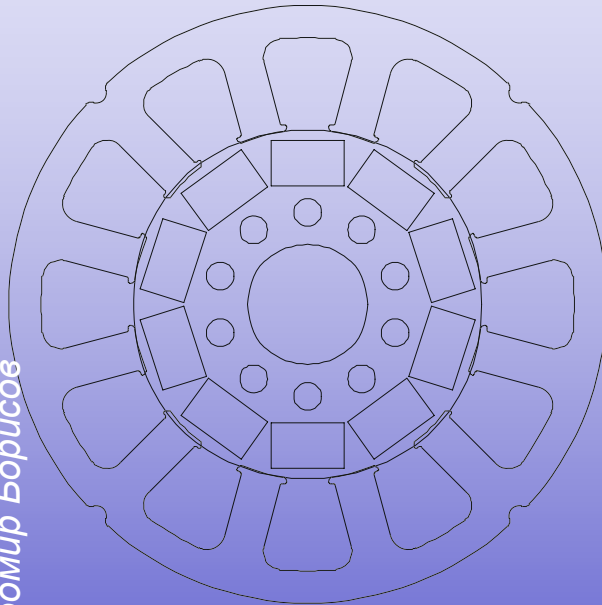
Любомир Борисов



Електрически двигатели. Основни конструкции и принципи.

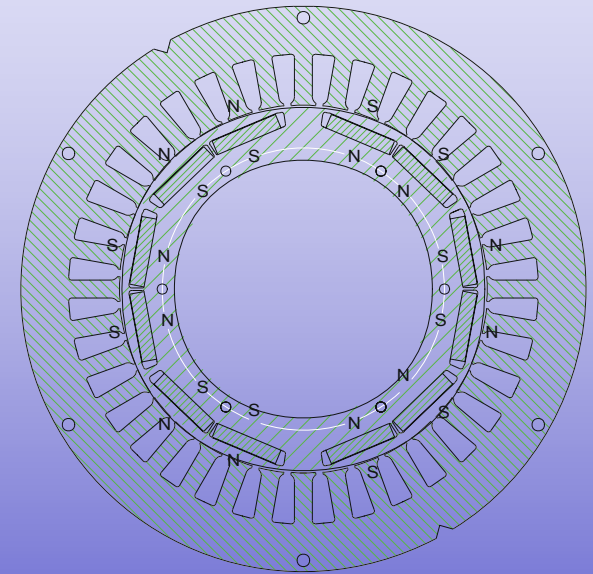
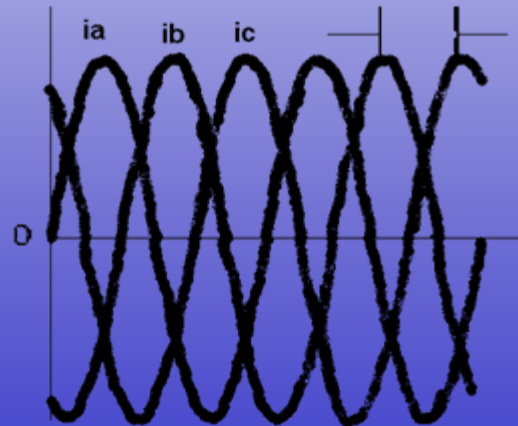
Синхронен електродвигател с постоянни магнити:

Любомир Борисов



3-phase

120°



Свойства на СМПМ:

Предимства:

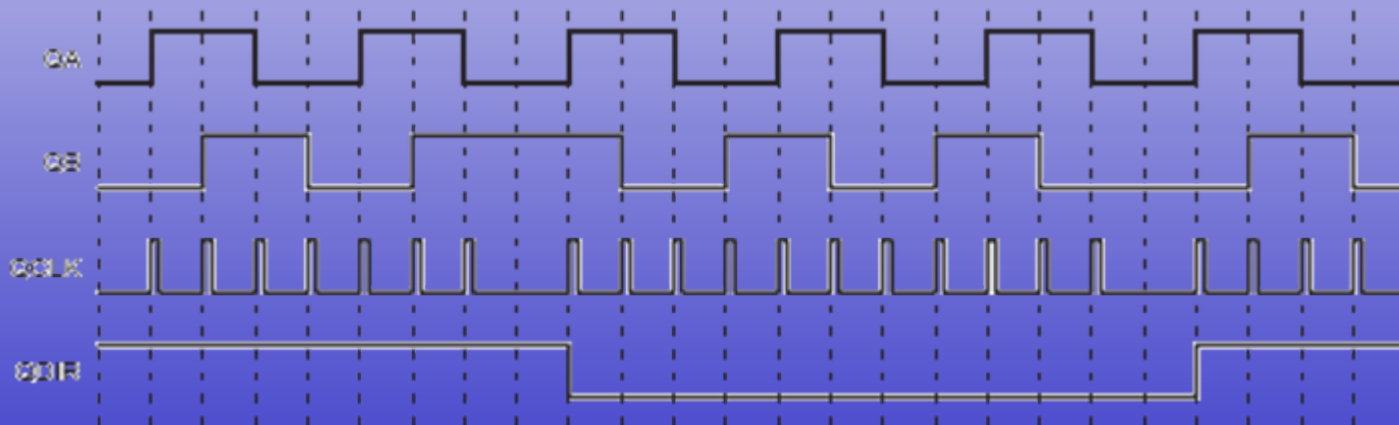
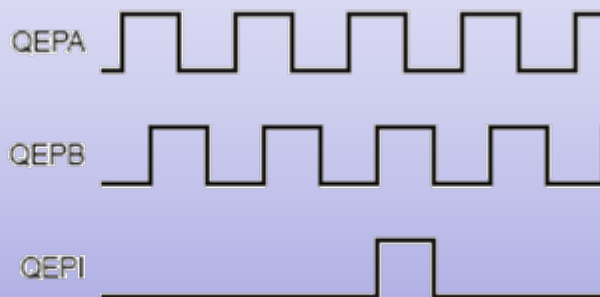
- Висок въртящ момент, граничещ с този на постояннотоковите мотори
- Отлични възможности за претоварване
- Добри регулировъчни свойства при използване на подходящи методи за управление
- Лесна поддръжка

Недостатък:

- „Cogging“ (EN), „Rastmoment“ (DE) – увеличаване на съпротивителния момент при преминаване под полюс.

Обратни връзки по позиция.

**Фоторастерови и фотокодови преобразуватели - ФРП
(QEP Encoders) и ФКП:**



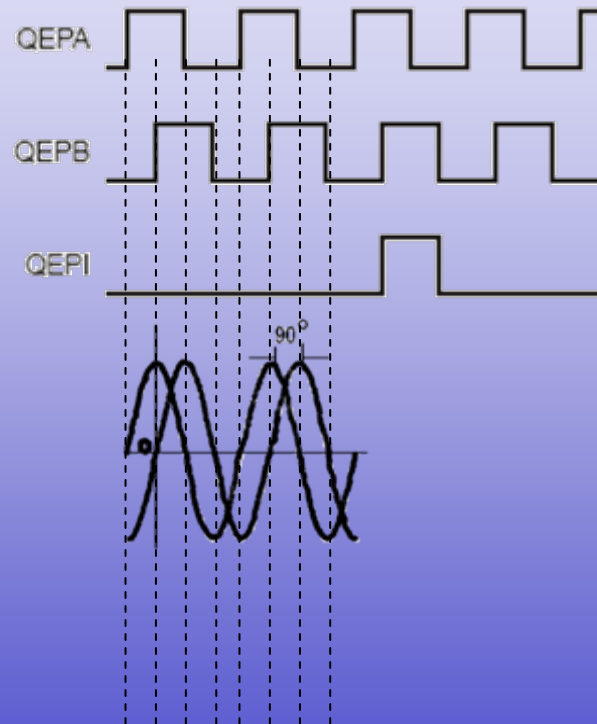
Любомир Борисов



Обратни връзки по позиция.

AMK
Control your Motion.
PLControl

Синусоидални енкодери:



Любомир Борисов



Катедра “Автоматика,
информационна и управляваща
техника”

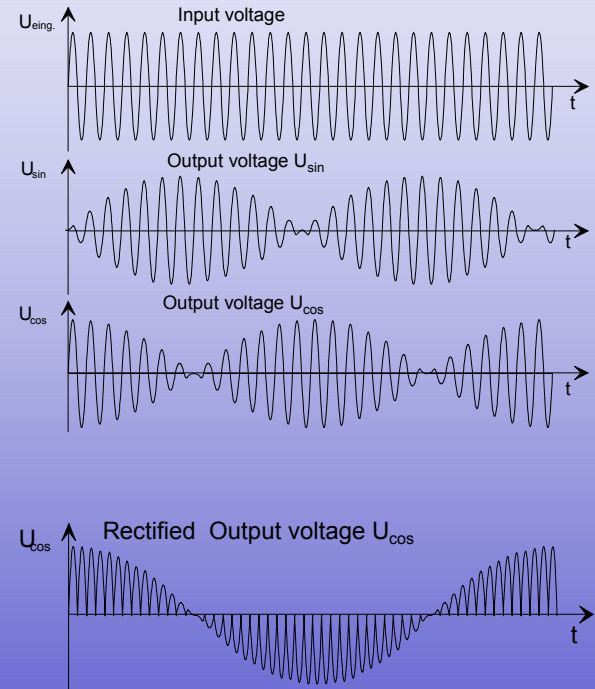
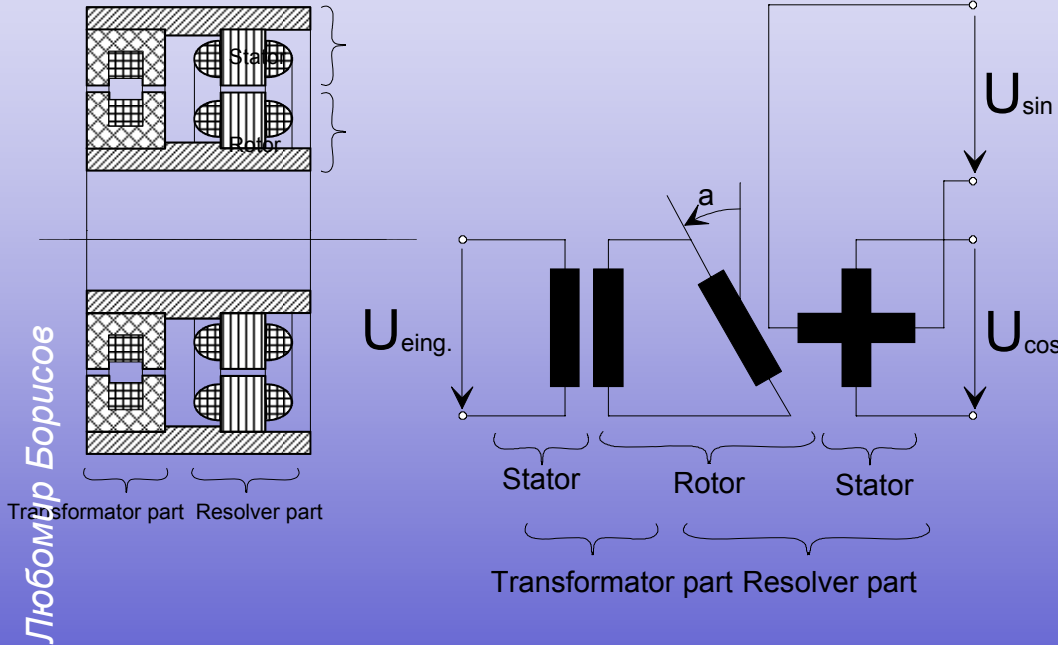
Обратни връзки по позиция.

AMK

Control your Motion.

PLControl

Резолвери:



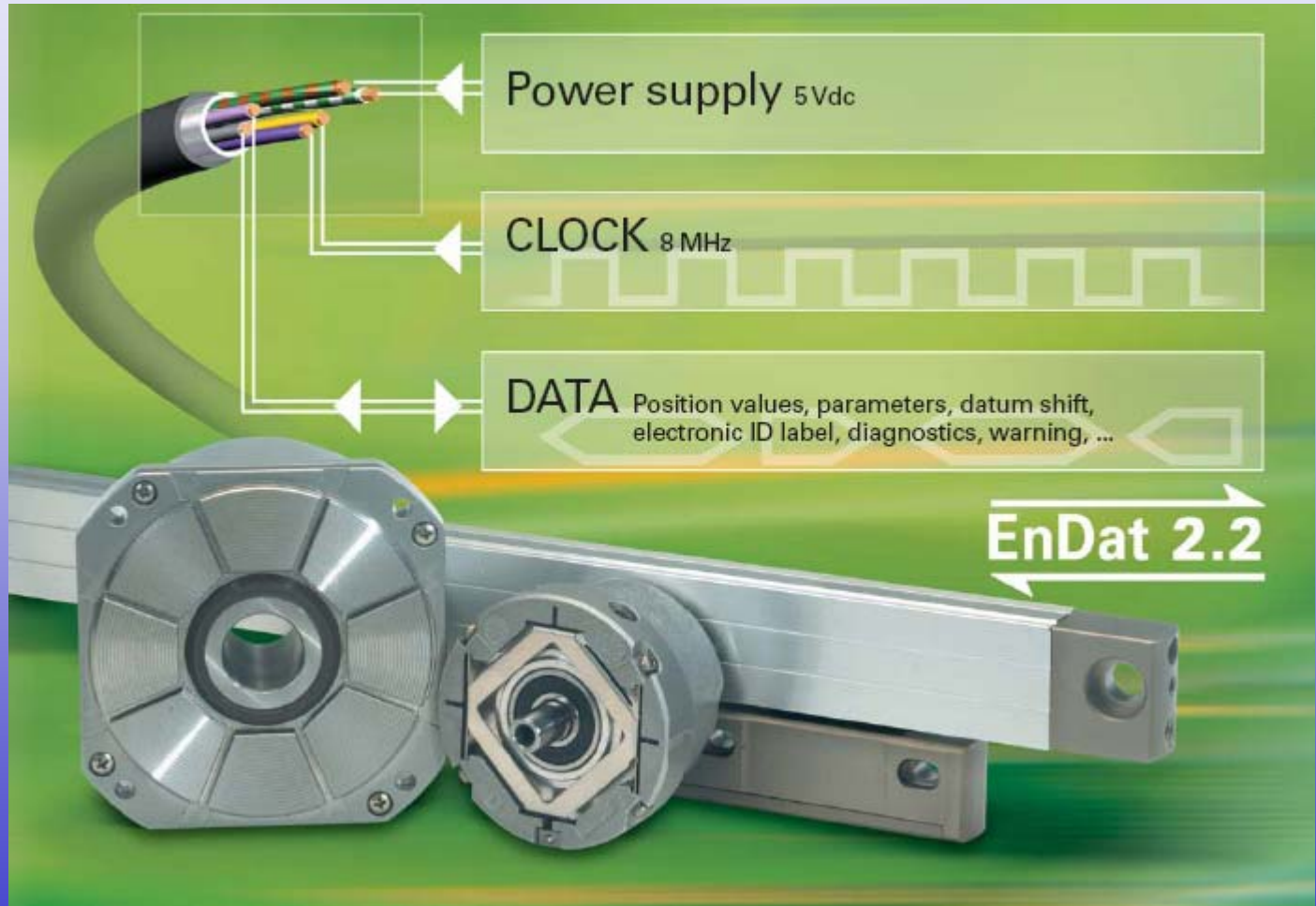
Обратни връзки по позиция.

AMK

Control your Motion.

PLControl

Енкодери с
цифров сериен
интерфейс:



Любомир Борисов



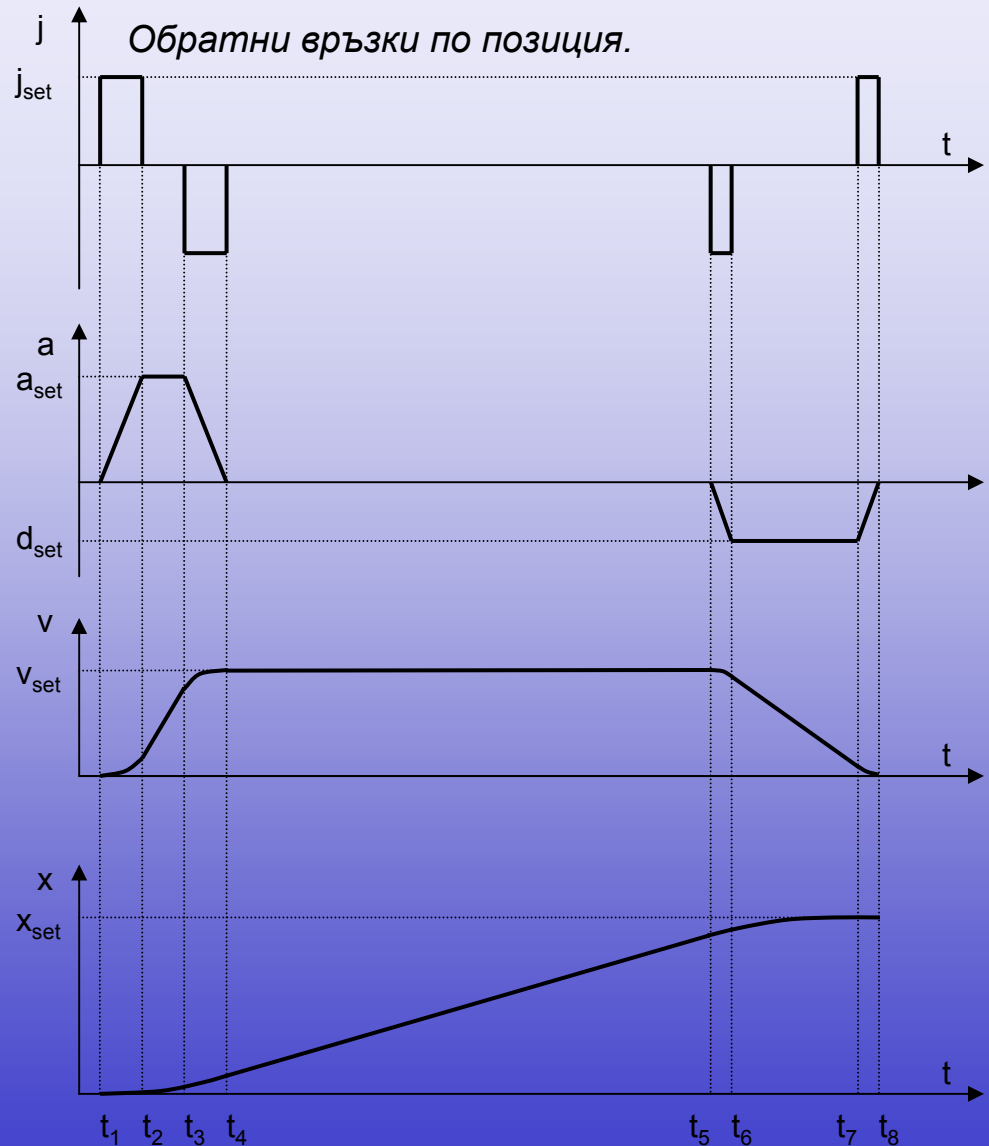
Катедра “Автоматика,
информационна и управляваща
техника”

Енкодери – сравнителна характеристика:

	Точност	Разрешаваща способност	Бързина на преобразуване	Изисквания към ап. част	цена
ФРП	~2 ' или по-лоша	До 10 000 [ipr]	Много висока - до $15 \cdot 10^7$ [cps]	Ниски	Средна
ФКП	~5 ' или по-лоша	До 4 096 [ipr]	Много висока - до $15 \cdot 10^7$ [cps]	Високи	Висока
Резолвери	~7 ' или по-лоша	До 65 536 [ipr]	Обикновено $8-16 \cdot 10^3$ [cps]	Високи	Ниска
Син. Енкодери	Достига ~2,5 "	До 1 280 000 [ipr]	Обикновено $8-16 \cdot 10^3$ [cps]	Средни	Много висока
Енкодери с цифров сериен интерфейс	Достига ~0.08 "	До 2^{24} [ipr]	Рядко над $8 \cdot 10^3$ [cps]	Ниски	Средна - до много висока
Прецизни Капацитивни	Достига 50 [pm]	5 [pm]	Ниска	Високи	Много висока

Извличане на информация
за други параметри на
движението от ОВ по
позиция

Любомир Борисов



Модул I. Съставни елементи на електрозадвижващите системи

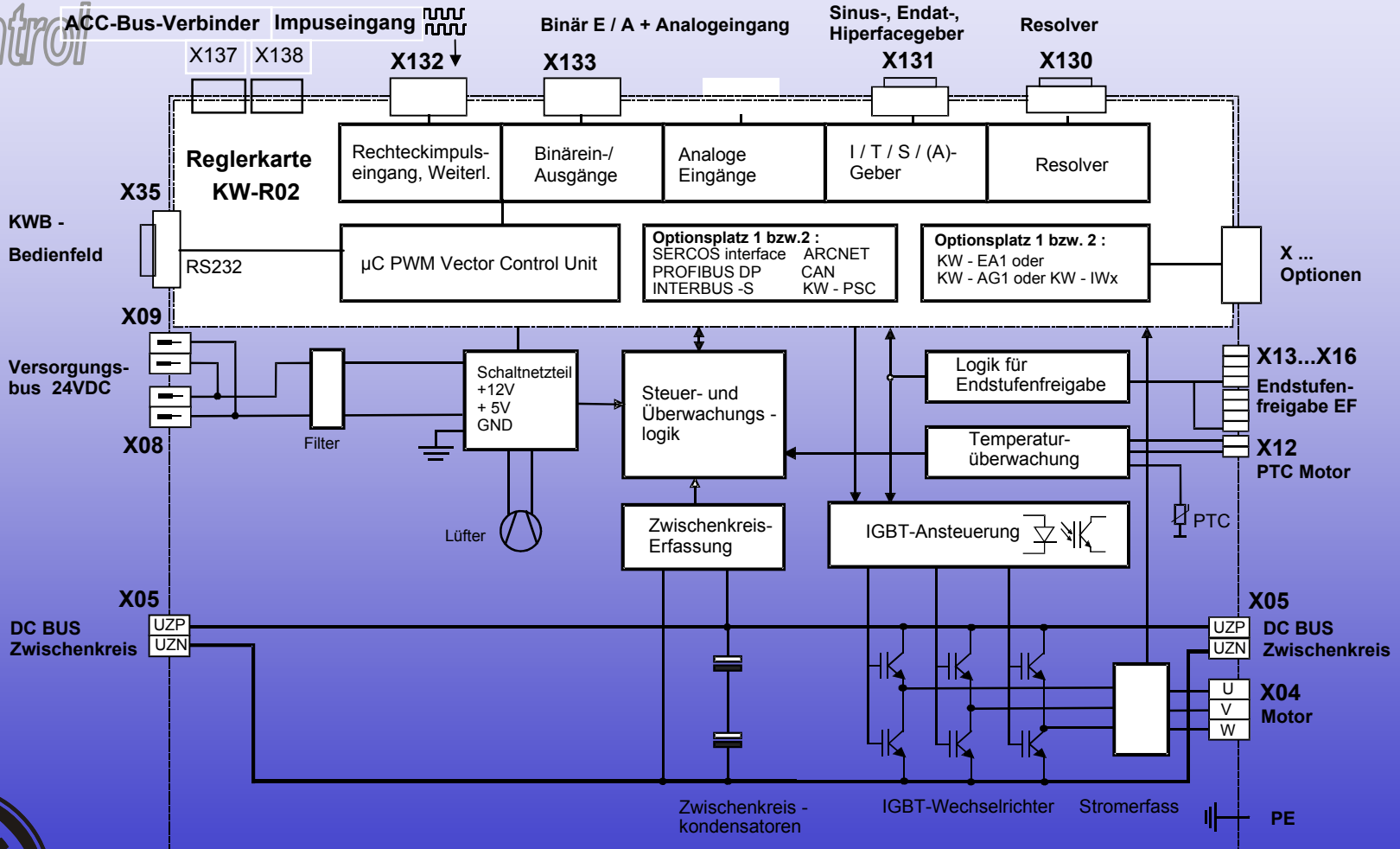
Честотни инвертори и серво-задвижвания

AMK
Control your Motion.

Серво

PLControl

Любомир Борисов

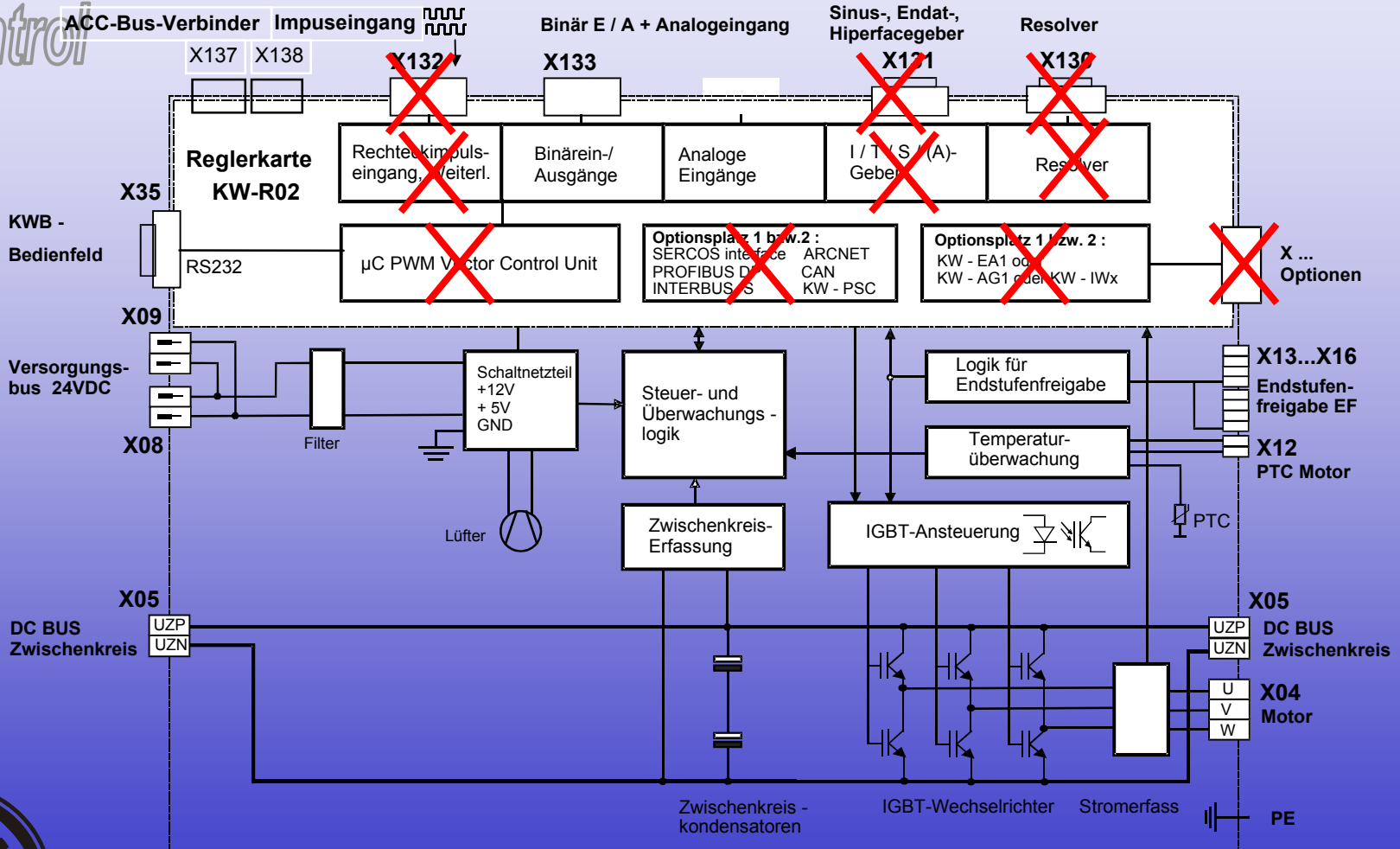


Катедра "Автоматика,
информационна и управляваща
техника"

АИУТ/КСУ Летен семестър 2008/2009

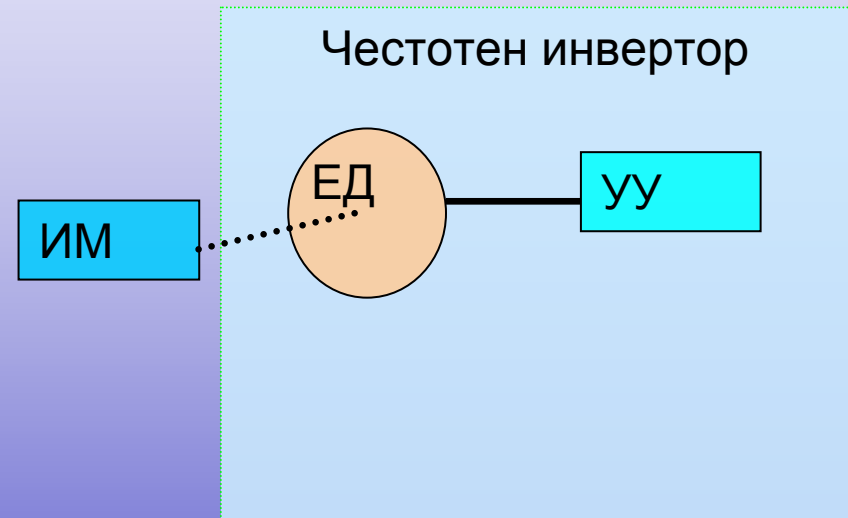
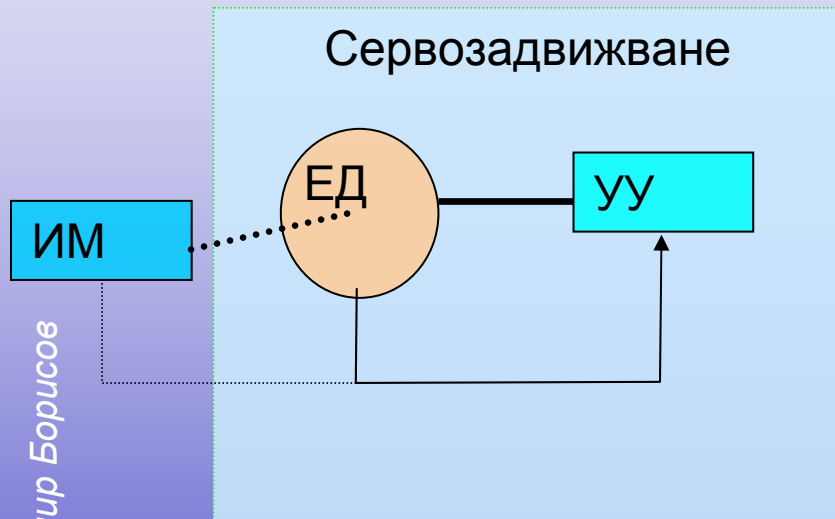
Честотно задвижване

PLControl



Любомир Борисов

Разлика между обикновен честотен инвертор и сервозадвижване



Любомир Борисов

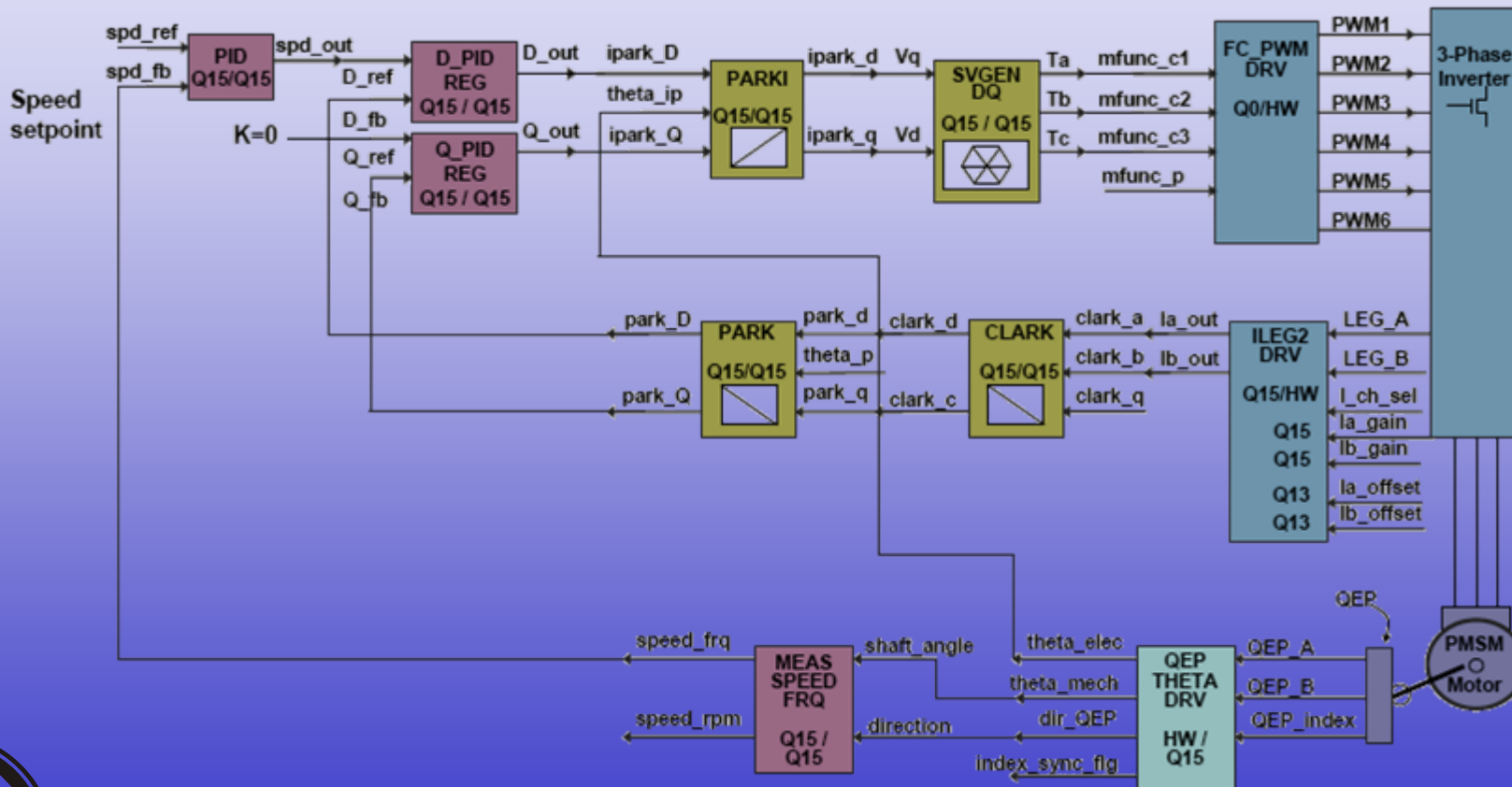




Control your Motion.

PLControl

Управление на СМГМ чрез ориентация на магнитното поле и пространствено-векторна модулация

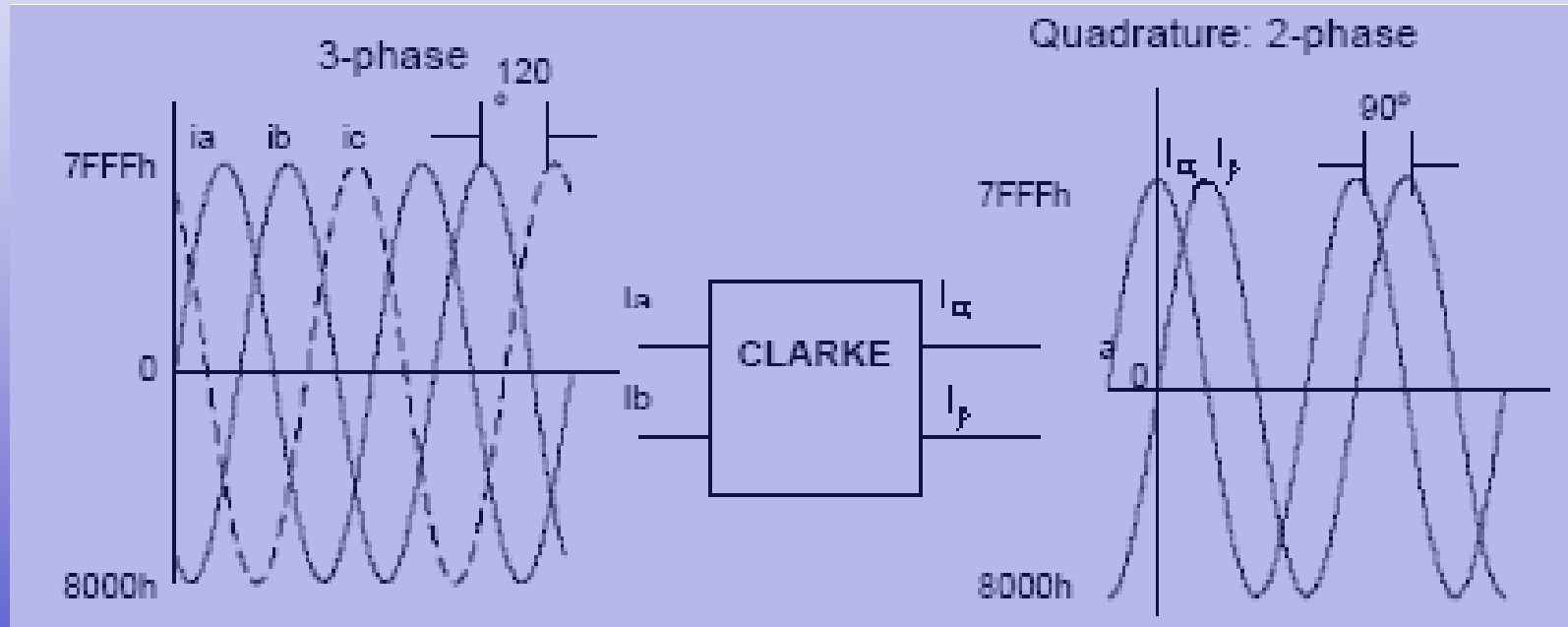


Любомир Борисов



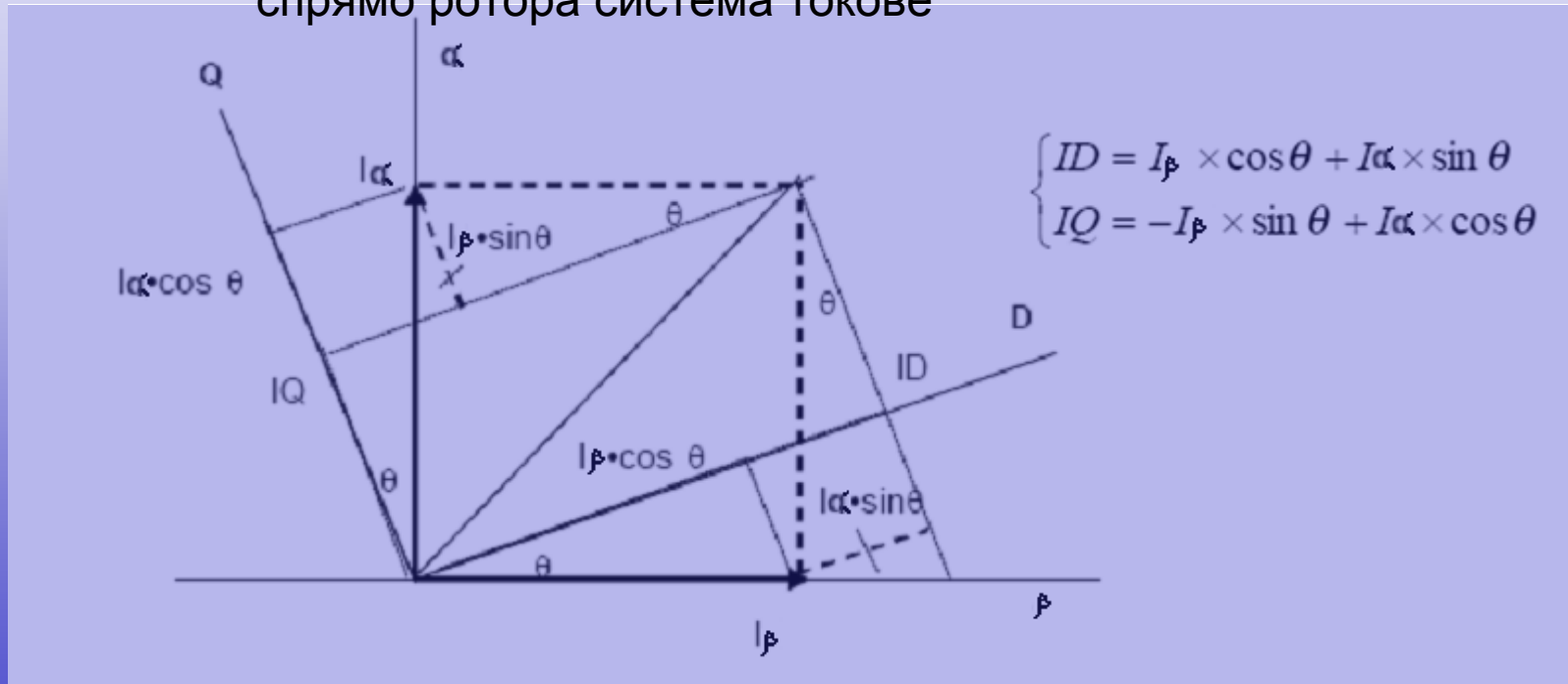
Clark-Трансформация

Преобразува симетрична трифазна система (токове) в симетрична двуфазна квадратична система



Трансформация на Park-Горев

Преобразува токовете вектори от неподвижната правоъгълна двуфазна координатна система на статора във въртяща се спрямо ротора система токове

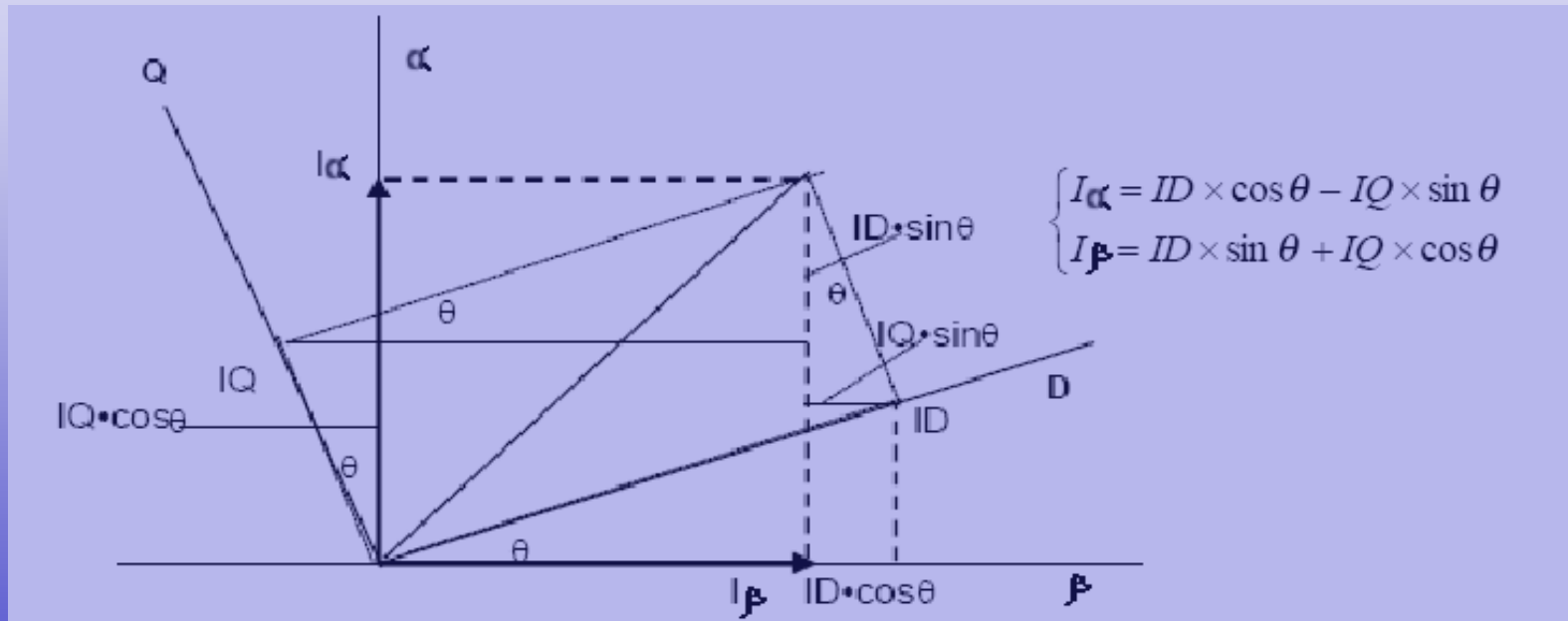


Любомир Борисов



Инверсна Park-Трансформация

Проектира системата роторни токове обратно в ортогонална за статора координатна система



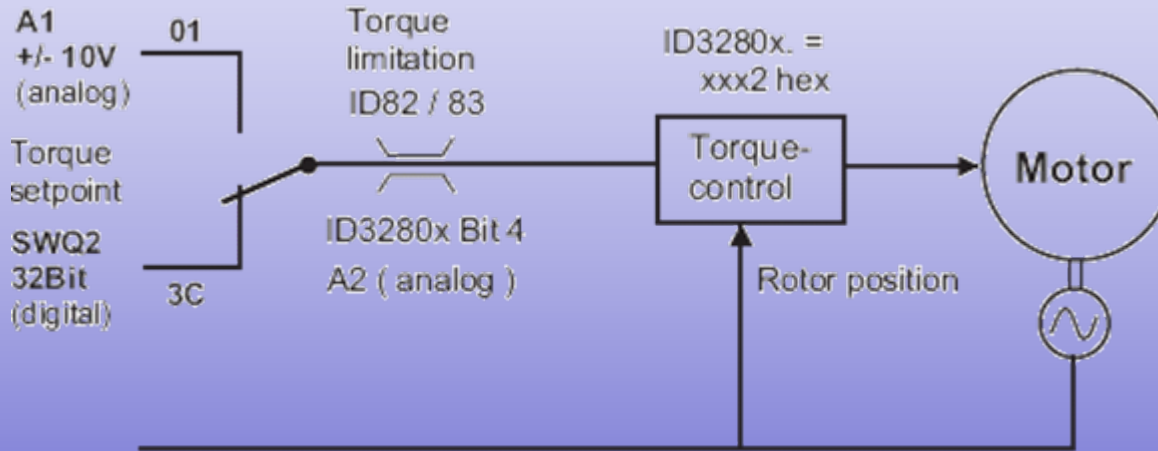
Любомир Борисов



Важни параметри при избор на електрозадвижваща система:

- Номинален и максимален въртящ момент, необходимост от претоварване
- Номинална и максимална скорост
- Точност на поддържане на скоростта и позицията
- Разрешаваща способност на енкодера
- Степен на защита (IP) на мотора, енкодера и управляващото устройство
- Разсейвана мощност
- Категория на безопасност
- Начин на охлаждане
- други

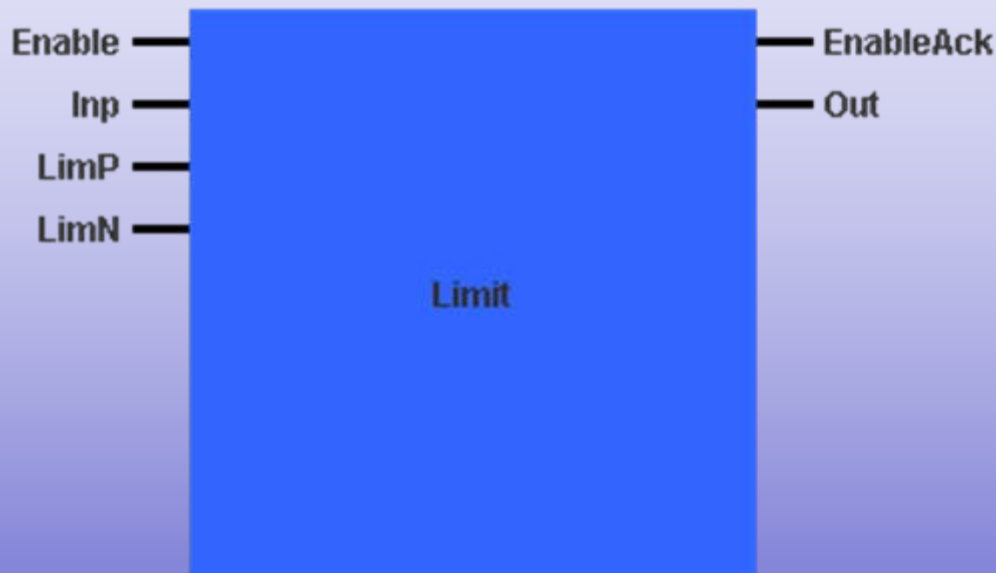
Контрол по въртящ момент



Любомир Борисов



Ограничител

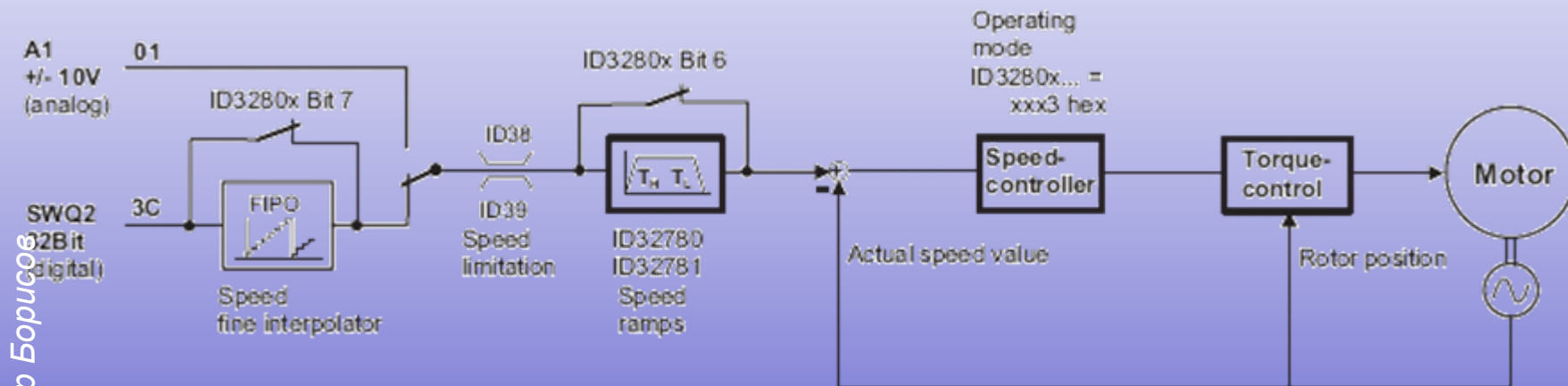


Out = Inp	LimP > Inp > LimN
Out = LimN	LimN ≥ Inp
Out = LimP	LimP ≤ Inp



PLControl

Контрол по скорост



Любомир Борисов

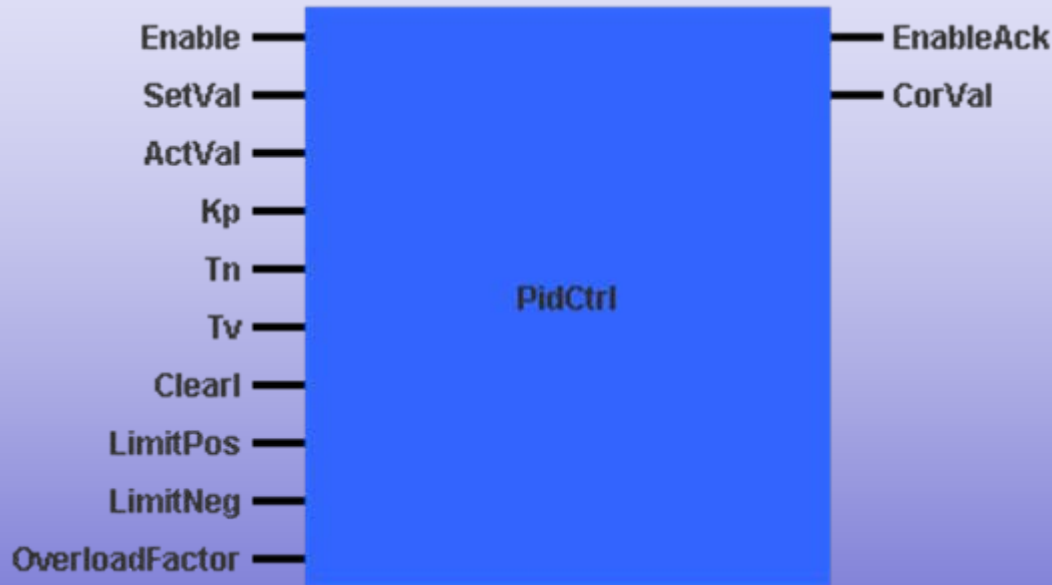




Control your Motion.

PLControl

PID с Anti-WindUp



$$\varepsilon(t) = SetVal(t) - ActVal(t)$$

$$CorVal(t) = Kp * \varepsilon(t) + \frac{Kp}{Tn} \sum_{n=1}^t \varepsilon(n) + KpTv[\varepsilon(t) - \varepsilon(t-1)]$$

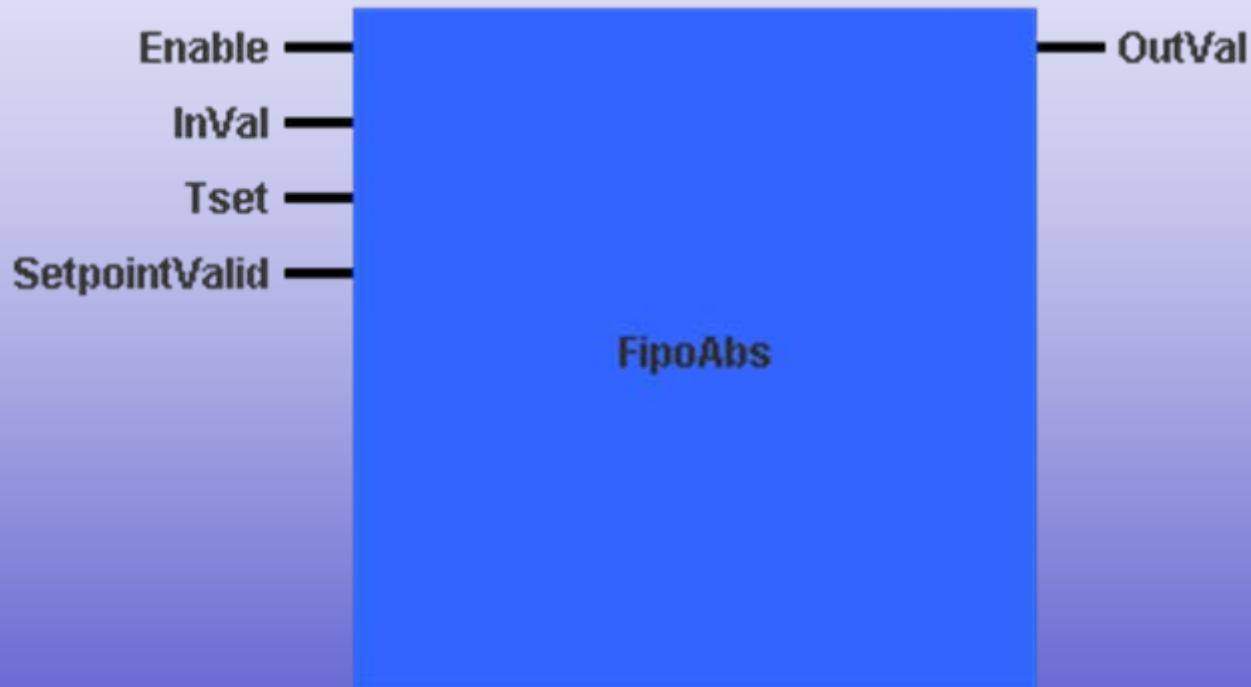
Любомир Борисов



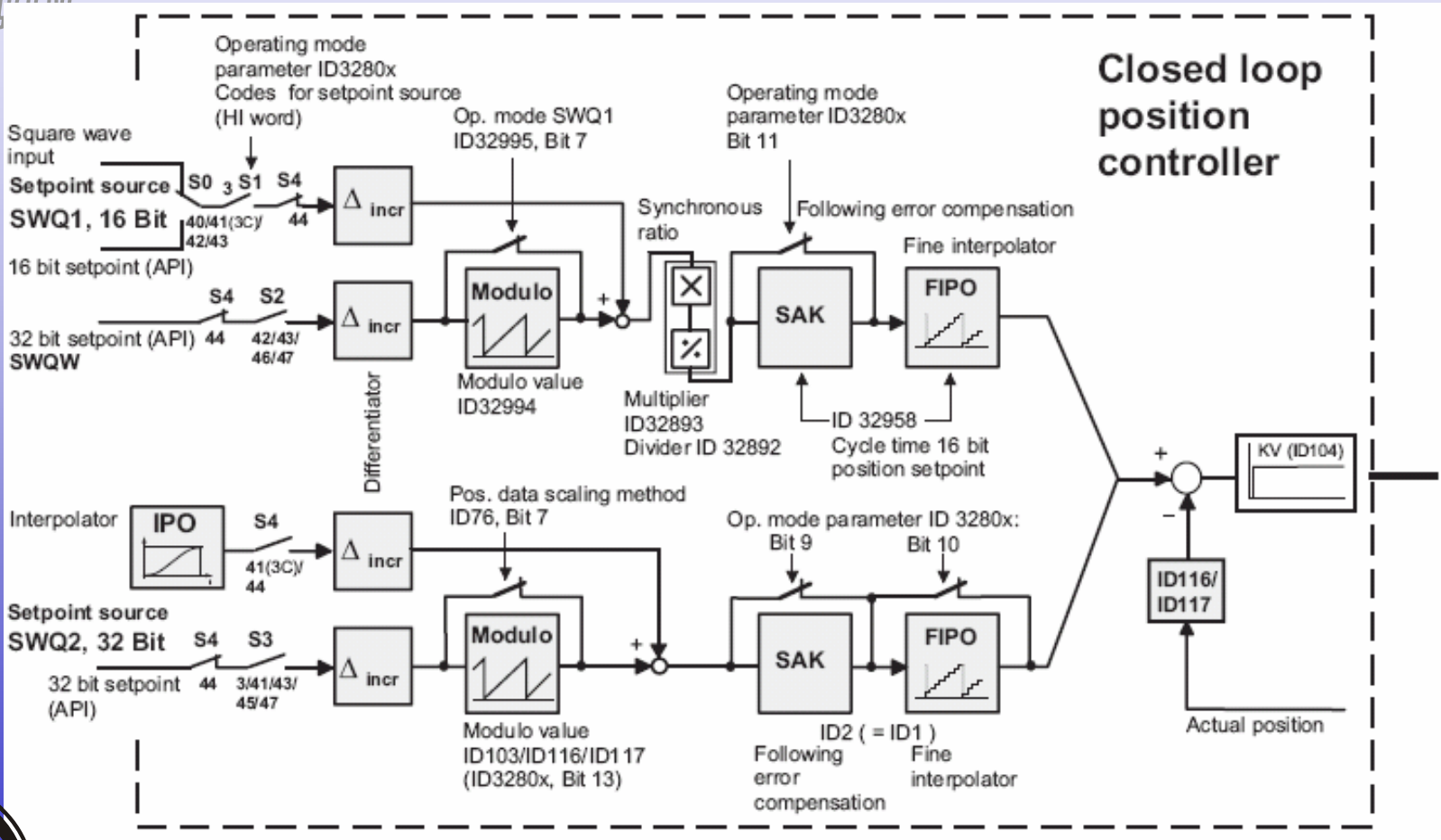
Рампи



Фин интерполатор



Контрол по позиция



Любомир Борисов



AMK

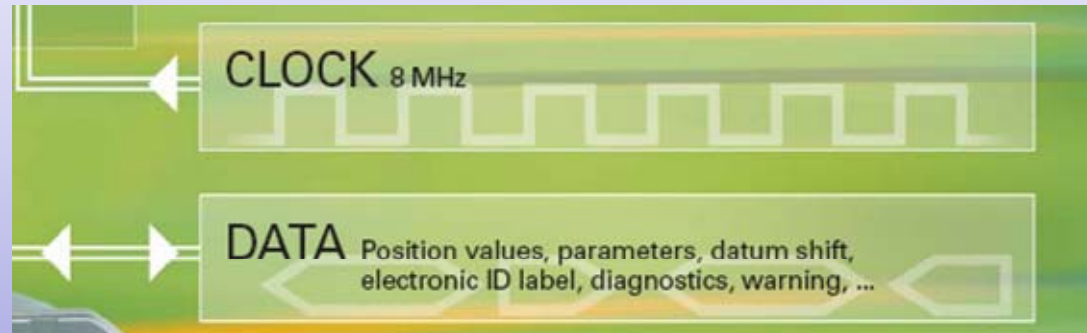
Control your Motion.

PLControl

Основни понятия. Серийна комуникация

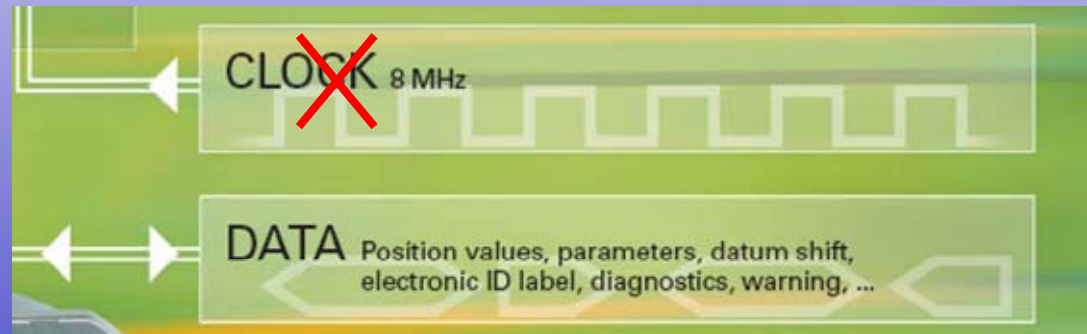
• Синхронна

Няма фиксирана дължина на бит, моментът на актуалност на състоянието се определя от фронт на тактовата поредица



• Асинхронна

Няма тактова поредица. Благодарение на фиксираната дължина на бит, моментът на актуалност на състоянието е ясно детерминиран.



Любомир Борисов



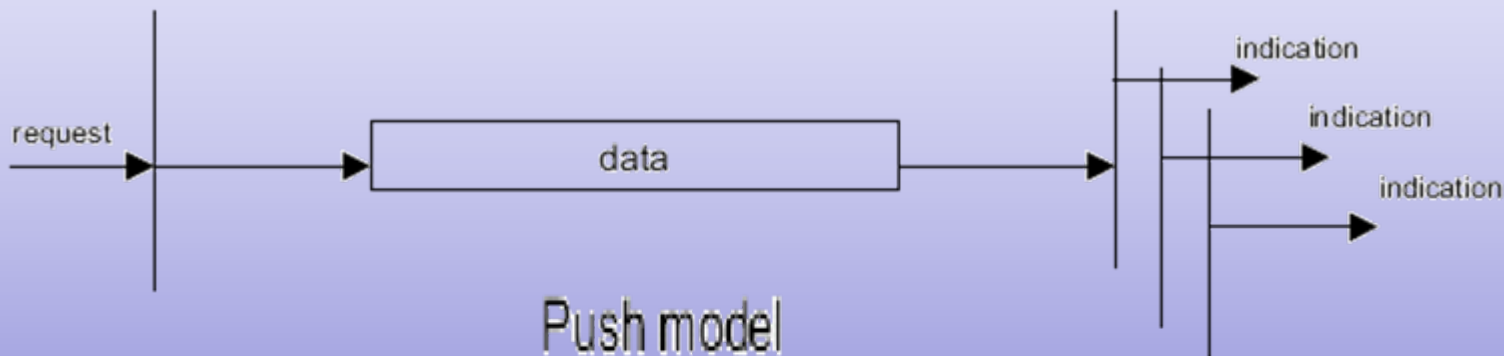
Катедра “Автоматика,
информационна и управляваща
техника”

Продуцент-консуматор

Основни понятия. Модели отношения между участниците.

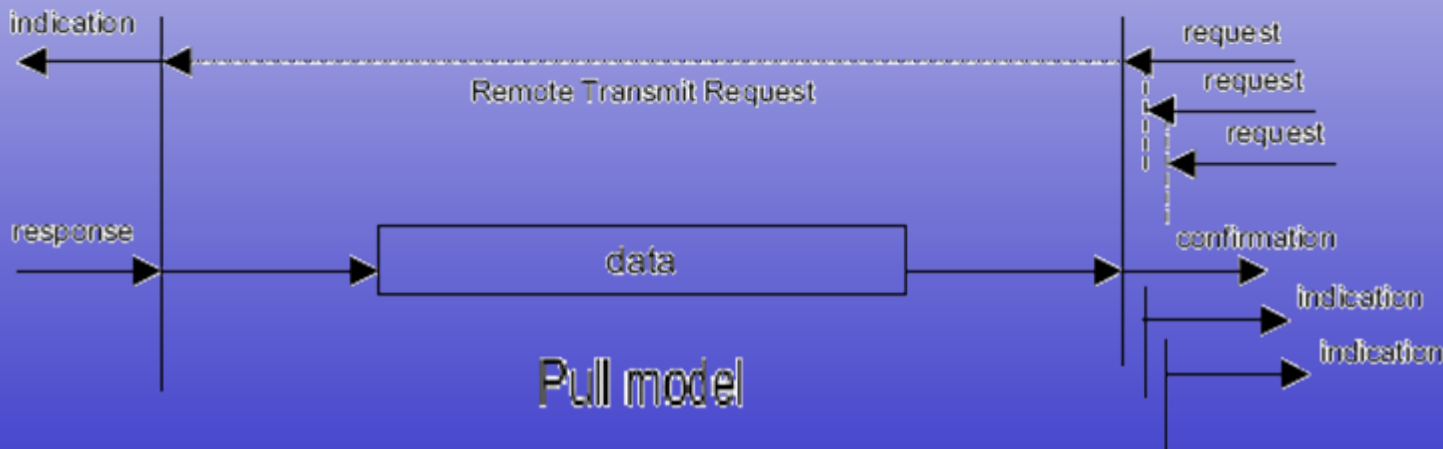
Producer

Consumers



Producer

Consumers

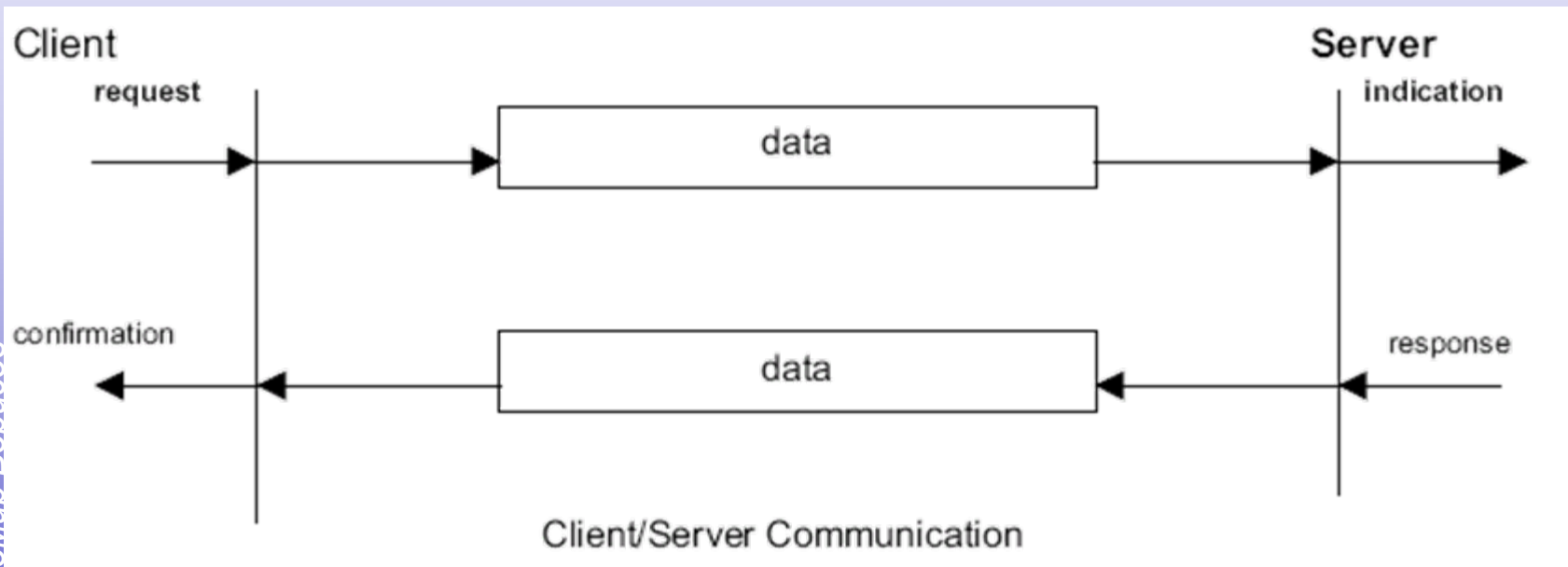


Любомир Борисов



Client-Server

Основни понятия. Модели отношения между участниците.

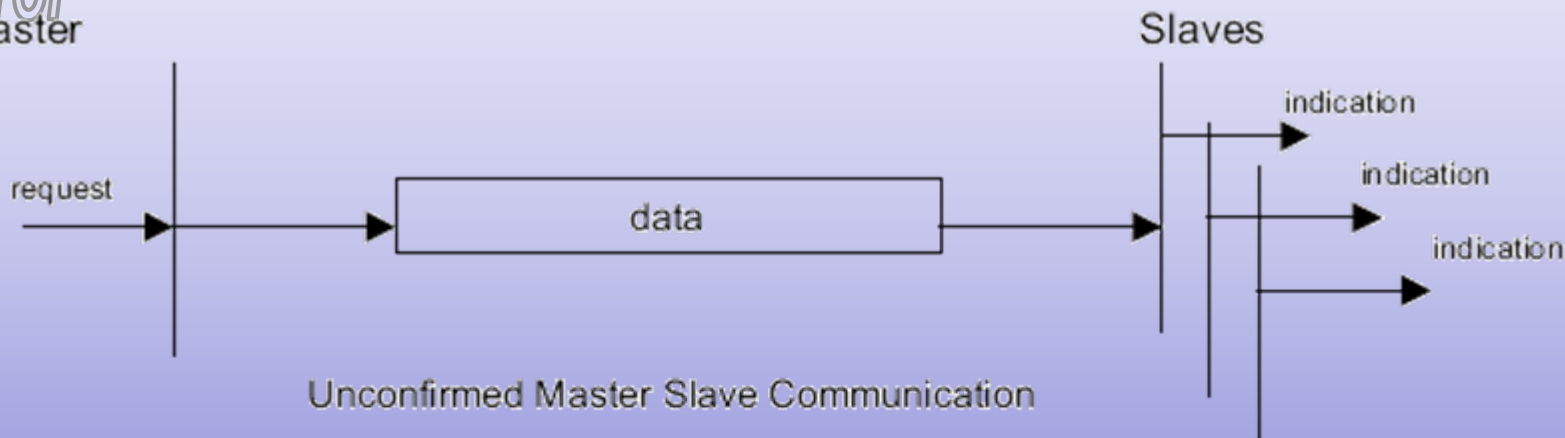


Любомир Борисов



Основни понятия. Модели отношения между участниците.

Master-Slave



Unconfirmed Master Slave Communication



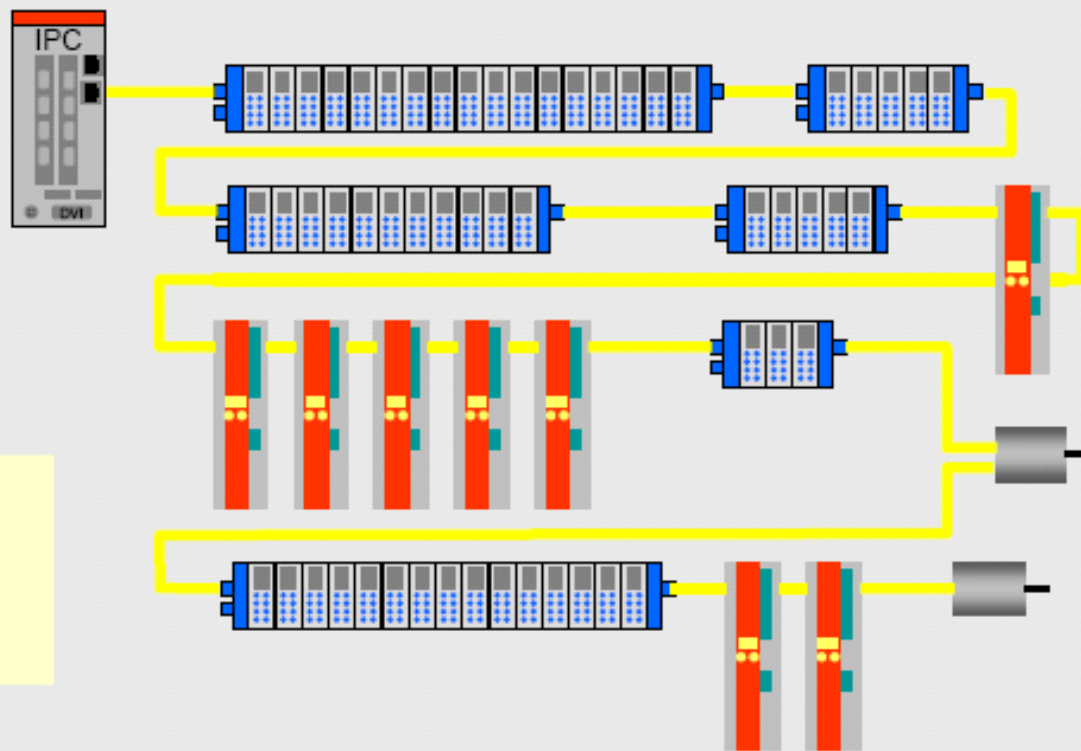
Confirmed Master Slave Communication

Любомир Борисов



Линейна

Line-Topology: any number of nodes lined up



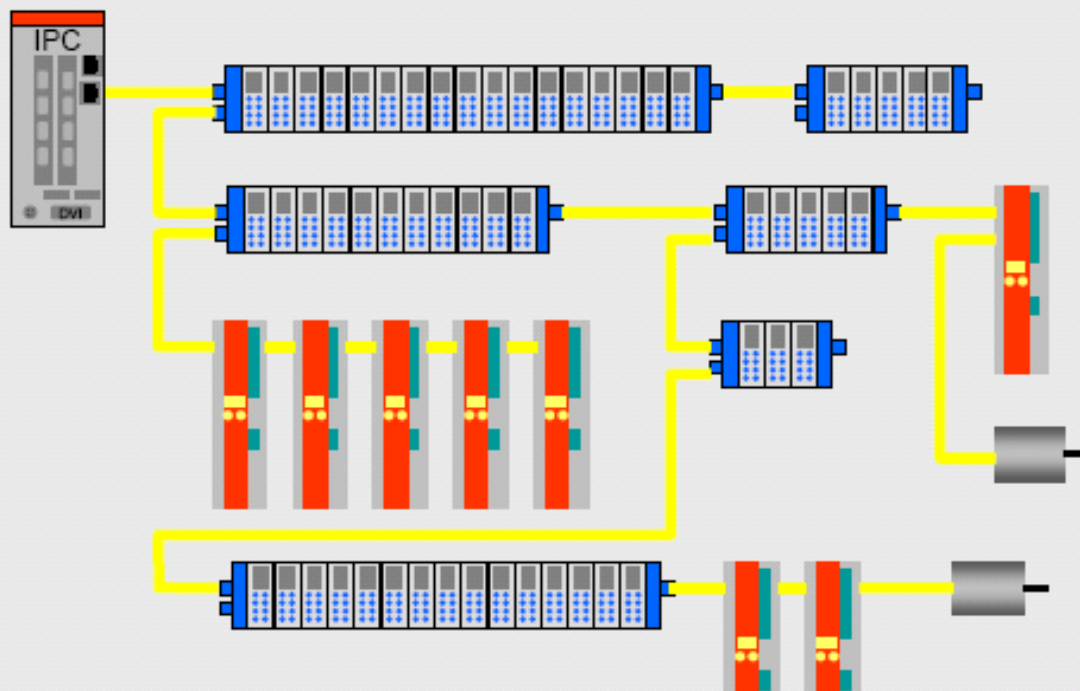
up to
65535
nodes

Любомир Борисов



Дървовидна

flexible tree structures – arbitrarily extendable



Любомир Борисов





Основни понятия

PLControl

Физическа среда

- RS485 – 5V диференциален сигнал – до 12 Mbps
- Оптични влакна – до 100 Mbps
- 10/100BASETX – до 100 Mbps
- LVDS – Low Voltage Differential Signal – 10Gigabit Ethernet.

Любомир Борисов



Катедра “Автоматика,
информационна и управляваща
техника”



Conventional NC drive technology

Основни понятия

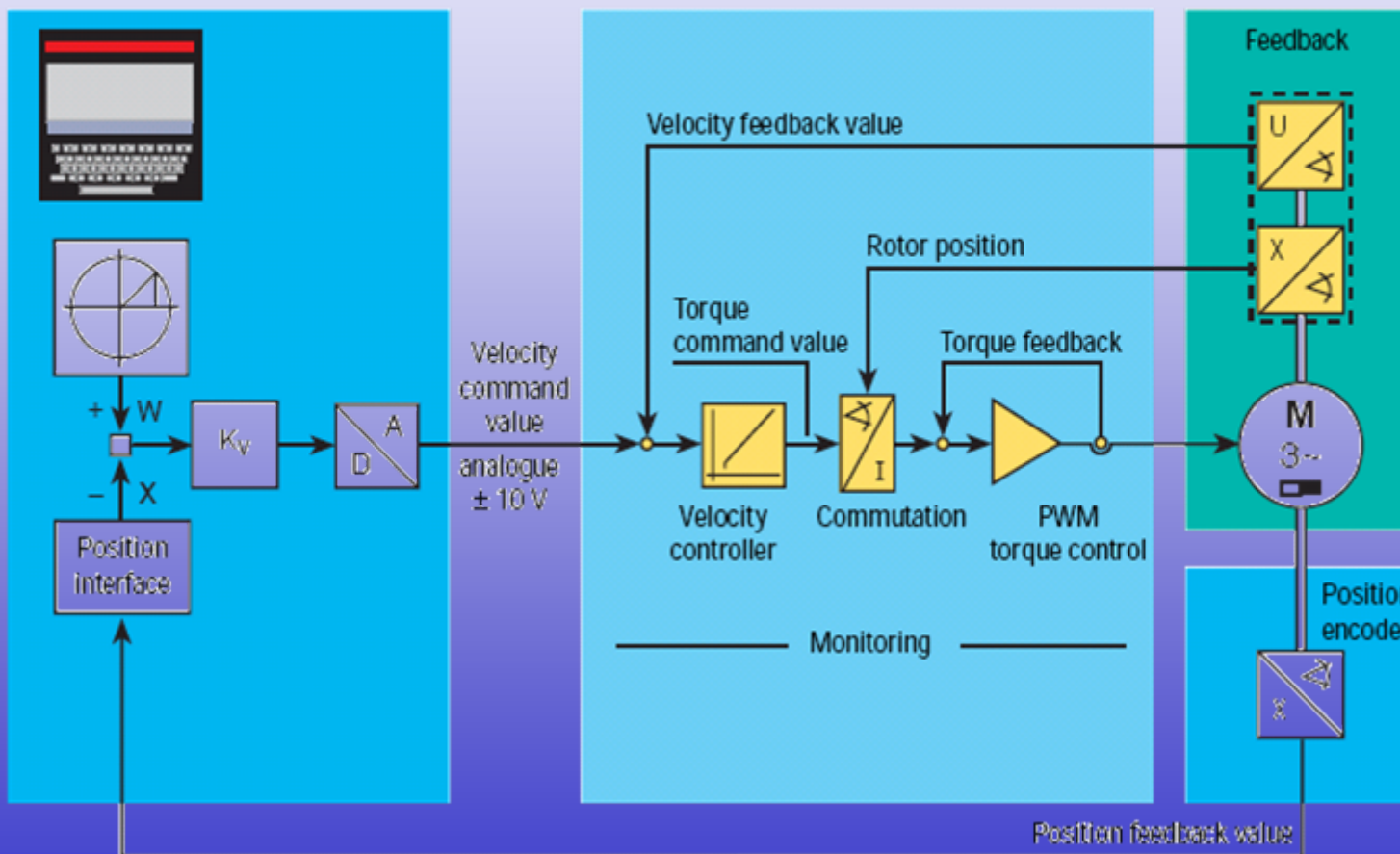
Control your Motion.

PLControl

Numerical control unit
(position control)

AC servo drive controller
(velocity control)

AC servomotor



Любомир Борисов



Катедра "Автоматика,
информационна и управляваща
техника"



Digital, intelligent NC drive technology

Основни понятия

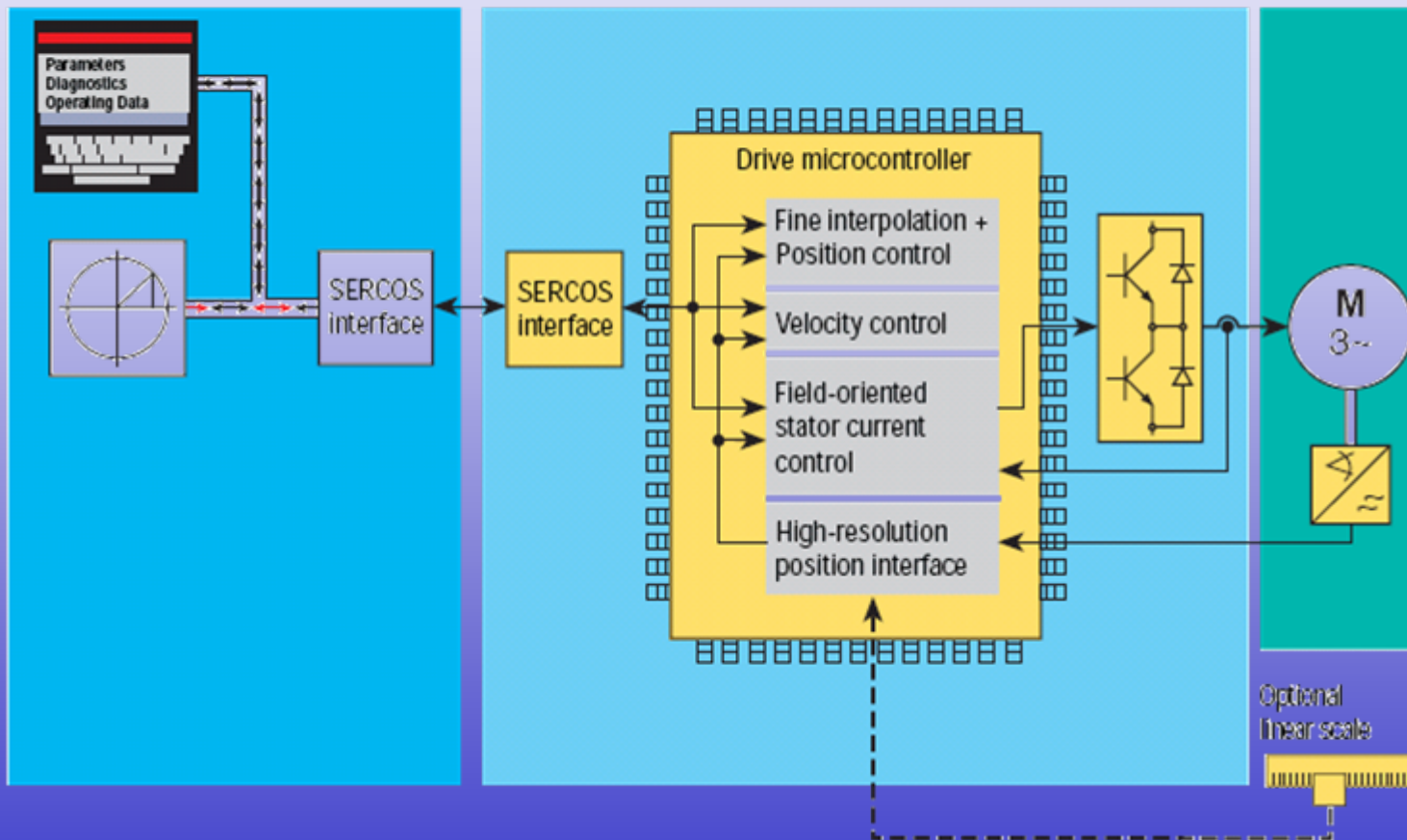
Control your Motion.

PLControl

Numerical control unit

Digital, intelligent AC drive controller with SERCOS interface

AC servomotor



Любомир Борисов



Катедра "Автоматика,
информационна и управляваща
техника"

АИУТ/КСУ Летен семестър 2008/2009



AMK

Control your Motion.

PLControl

Типични представители

- RS422 – 5V диференциален сигнал – до 12 Mbps
- RS232

PtP интерфейси

Любомир Борисов



Катедра “Автоматика,
информационна и управляваща
техника”

The logo for AMK, consisting of the letters 'AMK' in a bold, blue, sans-serif font.

Control your Motion.

PLControl

Свойства

SERCOS

- Осигурява цифрово предаване на заданията към задвижването и на обратните връзки в рамките на конкретен цикъл във всички устройства по мрежата. Продължителност на цикъла 62 μ s, 125 μ s, 250 μ s и всички кратни на 250 μ s до 65 ms;
- Синхронизация – за да се движат всички задвижвания синхронно трябва измерването на актуалната позиция да става едновременно с точност до микросекунда във всички устройства, а заданията да постъпват в един и същи цикъл;
- Параметризация и управление от контролни терминали;
- Скорост на предаване на данните 2, 4, 6, 8 или 16 Mbps при оптична среда на разпространение;
- До 254 участника в един оптичен кръг.

Любомир Борисов

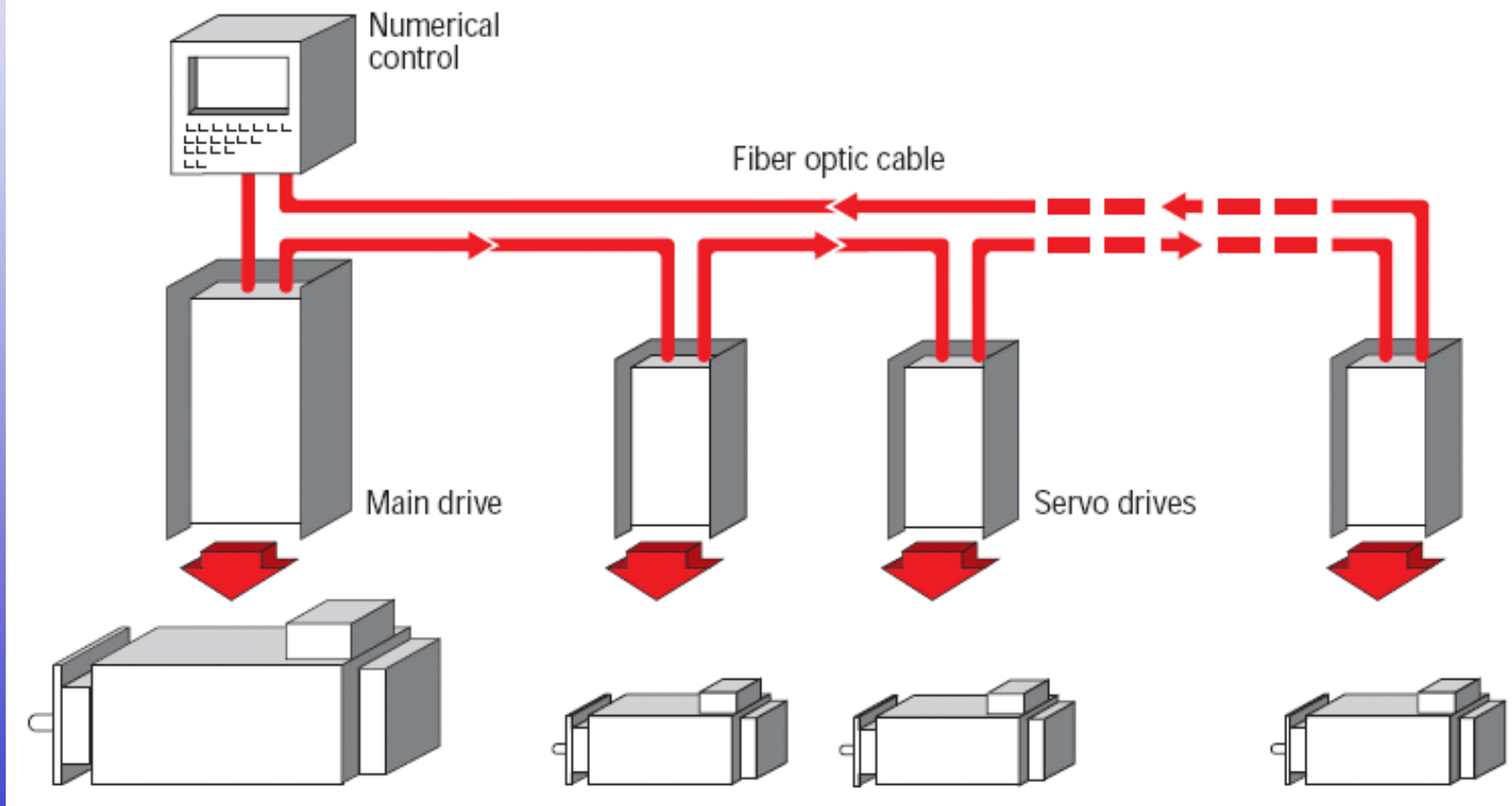


Катедра “Автоматика,
информационна и управляваща
техника”

SERCOS

Топология

Digital AC drives with the SERCOS interface

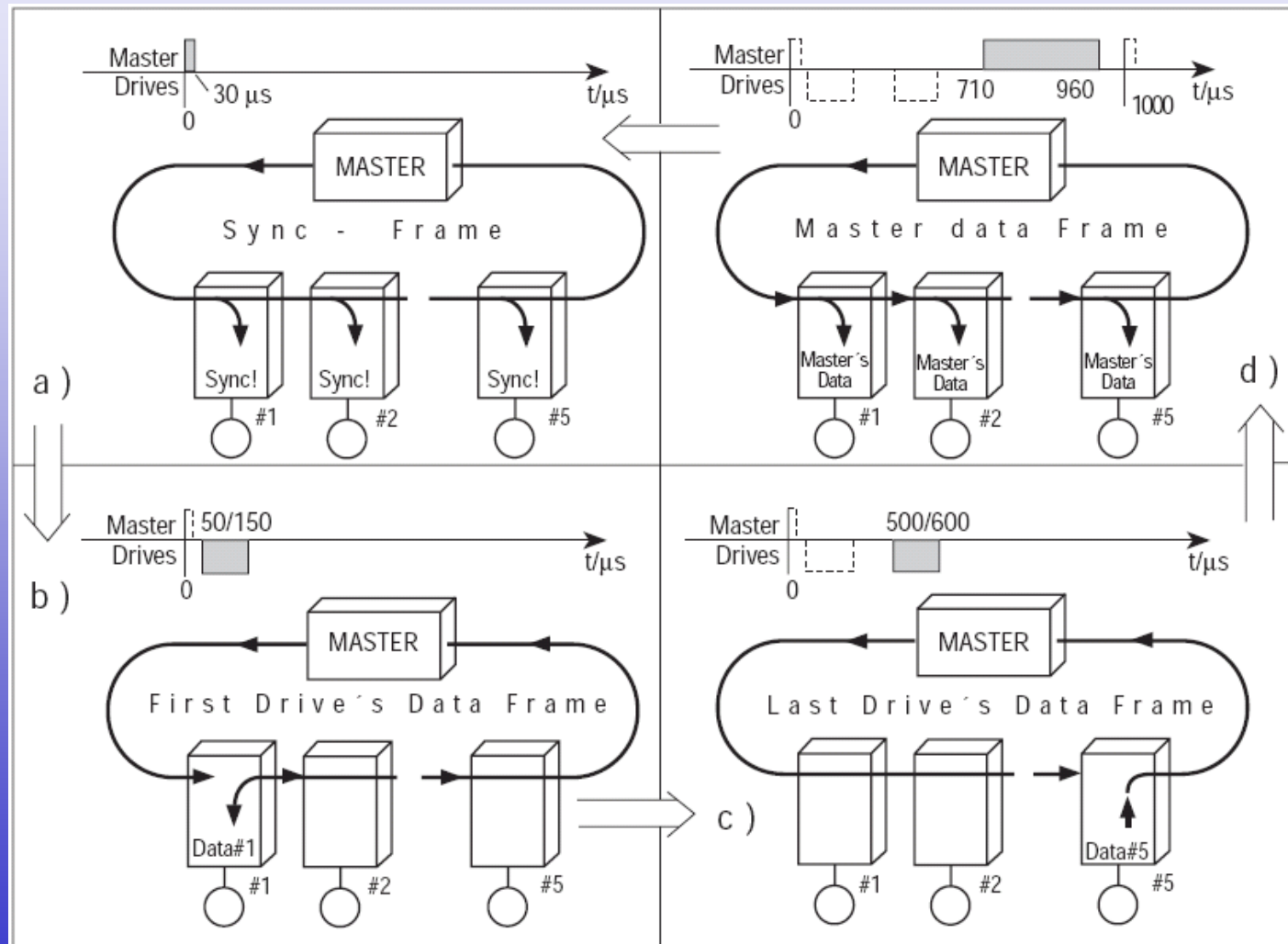


Любомир Борисов



SERCOS

Структура на комуникацията

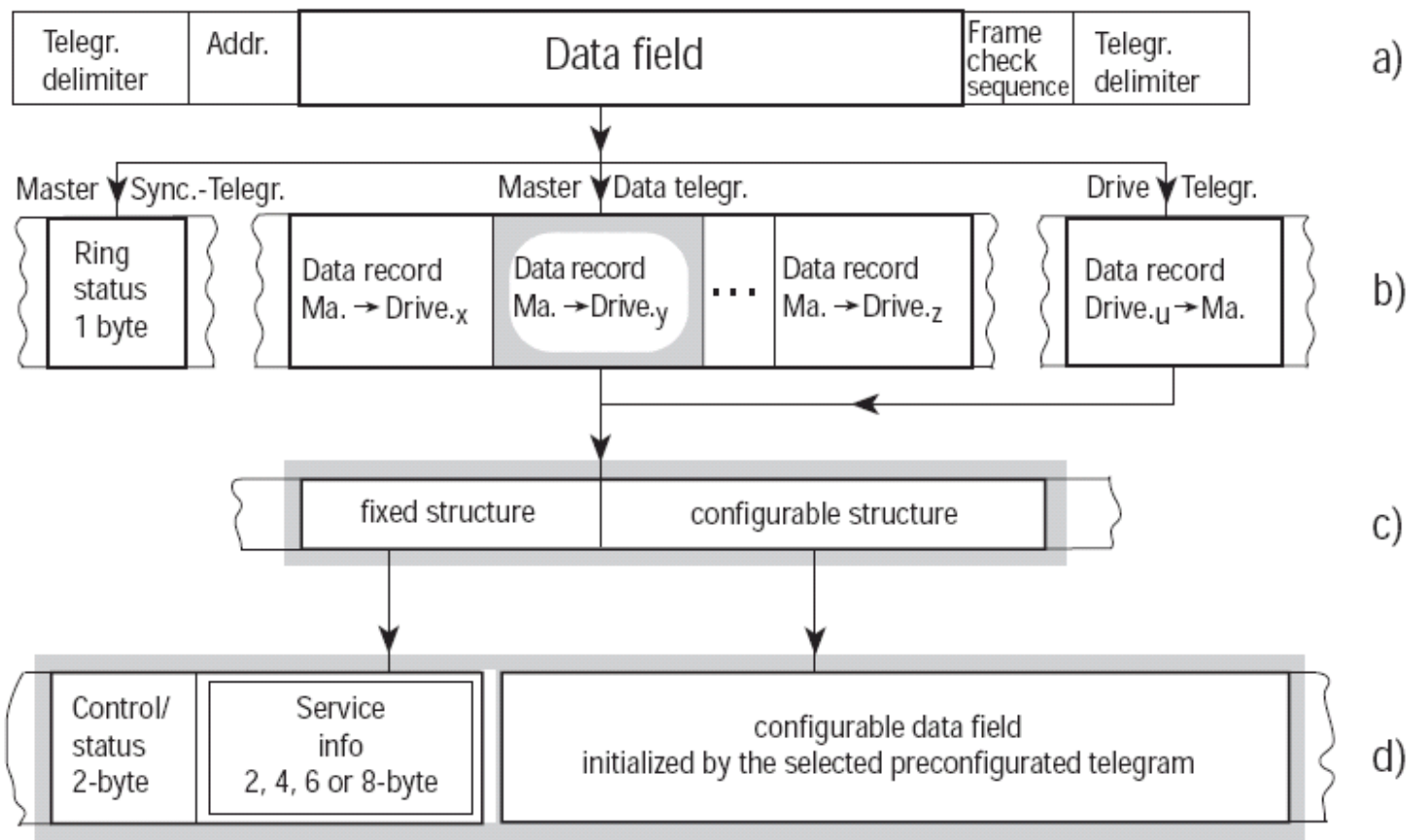


Любомир Борисов



SERCOS

Структура на телеграмата





Control your Motion.

Трансфер на параметри

PLControl

SERCOS

Excerpt from drive and control adaptation parameters

Controller-specific applications parameters		Machine-specific applications parameters	
IDN:	Function:	IDN:	Function:
S-0-0032	Primary mode of operation	S-0-0041	Homing velocity
S-0-0033	Secondary operation mode 1	S-0-0042	Homing acceleration
S-0-0034	Secondary operation mode 2	S-0-0049	Positive position limit value
S-0-0035	Secondary operation mode 3	S-0-0050	Negative position limit value
S-0-0044	Scaling of velocity data	S-0-0054	Actual position feedback 2
S-0-0076	Position data scaling type	S-0-0115	Position feedback type parameter
S-0-0079	Rotational position resolution	S-0-0116	Resolution of rotational feedback 1
S-0-0086	Scaling type for torque force data	S-0-0118	Resolution of the linear feedback
S-0-0160	Scaling type for acceleration data	S-0-0121	Input revolutions of load gear
		S-0-0122	Output revolutions of load gear
		S-0-0123	Feed constant
		S-0-0147	Homing parameter
		S-0-0151	Reference offset 2
		S-0-0165	Distance coded reference dimension 1
		S-0-0166	Distance coded reference dimension 2

Любомир Борисов





AMK

Control your Motion.

PLControl

Profibus

PROFIBUS
IEC 61158 и IEC 61784

Варианти

- PROFIBUS-FMS (Fieldbus Message Specification) (1987)
Сложен и комплексен протокол
- PROFIBUS-DP (Decentralized Peripherals) (1993)
По-просто организиран и значително по-бърз протокол (DP-V0)
- PROFIBUS-PA (Process Automation)
Изискванията на процесната автоматизация включват механизми за увеличаване на сигурността

Физическа среда

- RS485
- RS485-IS – посреща изискванията за сигурност на експлоатацията
- MBP - "Manchester Coding(M)", (Bus Powering, BP)
- Оптични влакна

Любомир Борисов



Катедра "Автоматика,
информационна и управляваща
техника"



PL Control	MBP	RS485	RS485-IS	Fiber Optic / LWL
Среда за пренос	Цифров, Код на Манчестър	Цифров, диференциални сигнали RS485	Цифров, диференциални сигнали RS485	Оптичен, цифров
Скорост	31,25 KBit/s	9,6 до 12000 KBit/s	9,6 до 1500 KBit/s	9,6 до 12000 KBit/s
Сигурност на данните	Preamble, Защитен с/у грешки Старт/Стоп разделител	Parity bit, Start-и End-Delimiter	Paritybit, Start-и End-Delimiter	Paritybit, Start-и End-Delimiter
кабел	Усукана двойка	Усукана двойка	Усукана двойка	Стъклени или пластмасови влакна
Електрозахранване	Възможност за захранване по сигналните проводници	Възможно само по допълнителни проводници	Възможно само по допълнителни проводници	Възможно при използване на хибриден кабел
Огнеустойчивост	да	не	да	не
Топология	Линейна или дървовидна, комбинирана, с терминация	Линейна с терминация	Линейна с терминация	Звезда или кръг са типични, линейна е възможна
Брой на комуникационните възли	до 32 участника в сегмент; общо max. 126 за мрежата	до 32 участника в сегмент; общо max. 126 за мрежата с усилватели	до 32 участника в сегмент; общо max. 126 за мрежата	max. 126 за мрежата





AMK

Control your Motion.

PLControl

Profibus

Архитектура

Master-Slave архитектура.

Устройствата образуват „Token-Ring“. Устройството, в което се намира “Token”-а играе ролята на Master и инициира Комуникацията

Multi-Master система

Любомир Борисов





Развитие на DP интерфейса

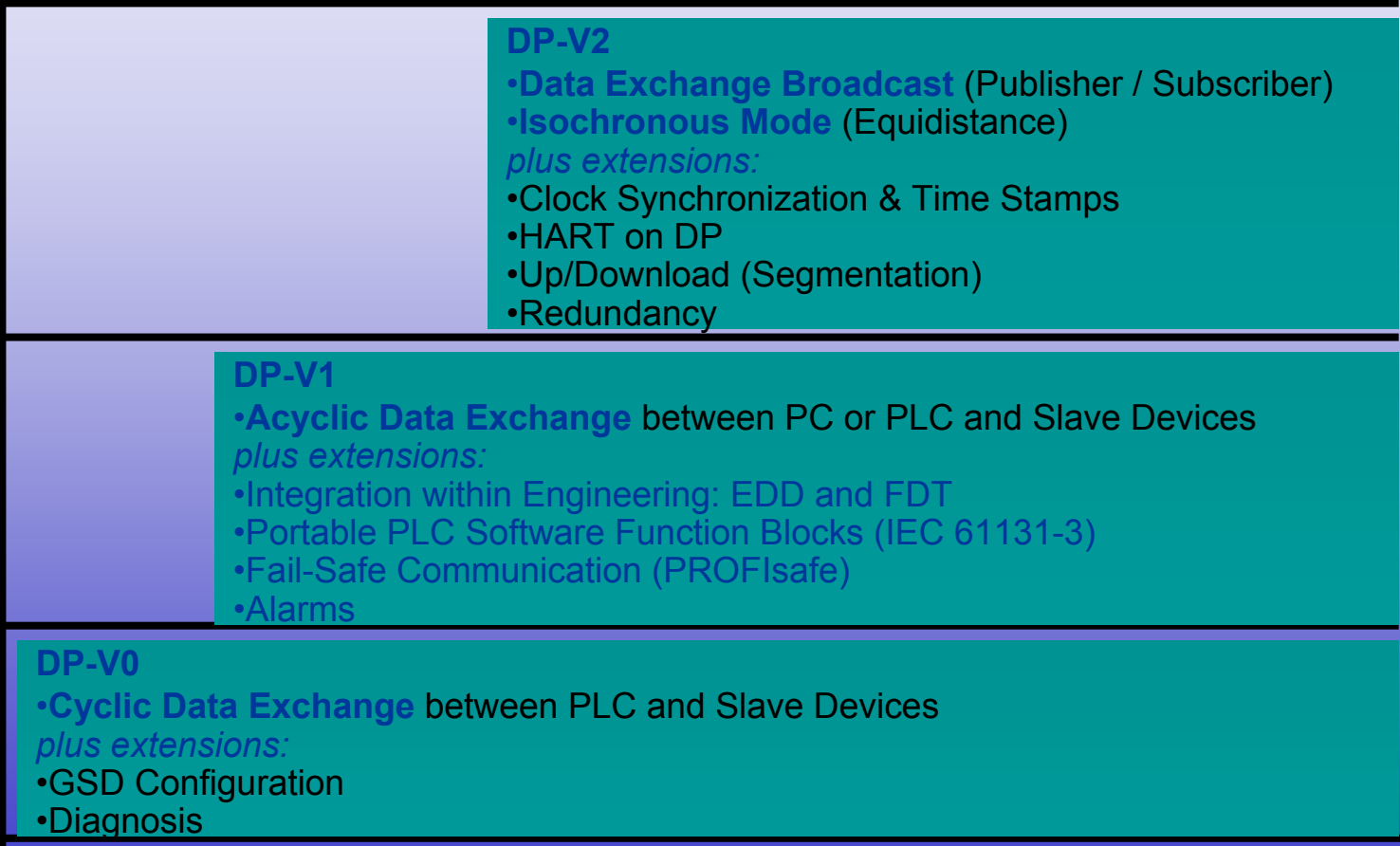
Profibus

PLControl

Features

Любомир Борисов

Device Features





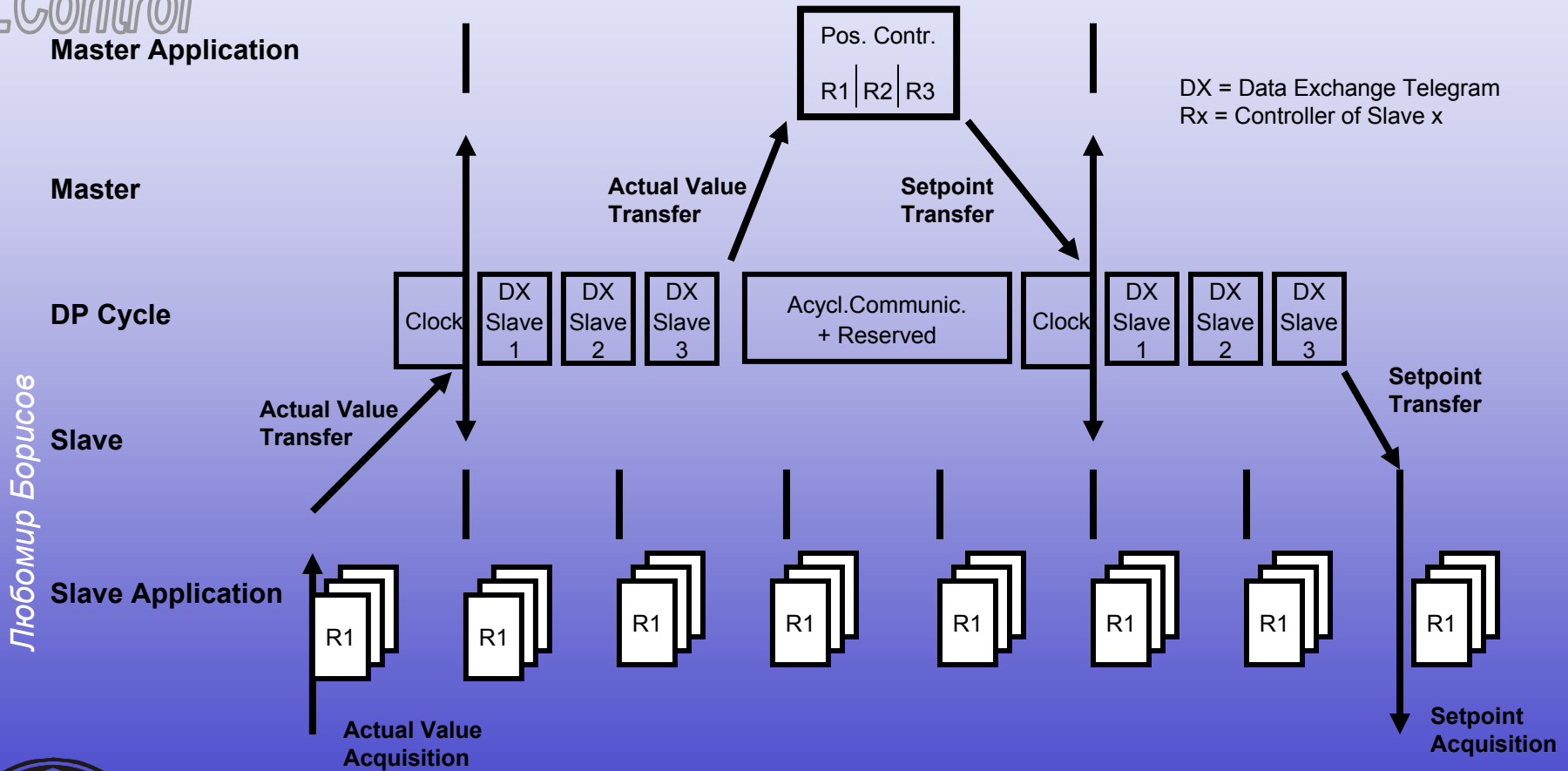
Control your Motion.

PLControl

Master Application

Profibus

Структура на комуникацията



Любомир Борисов



AMK

Control your Motion.

PLControl

Modbus:

Сериен комуникационен протокол създаден от Modicon през 1979г., с основно предназначение – комуникация между програмируеми логически контролери (PLC).

Предимства:

- отворен (безплатен) – спецификацията му е достъпна за всеки;
- Master/Slave – запитване/отговор;
- пренася данни без значение какви са;
- голяма популярност – може би е един от най-използваните протоколи в света.

Симеон Трифонов



Разновидности на Modbus:

- Modbus RTU: Най-разпространеният вариант. Използва интерфейс RS-232 или RS-485. Данните са в шестнадесетичен формат. Допуска само един Master.
- Modbus ASCII: Също използва RS-232 или RS-485. Данните са в ASCII формат. Допуска само един Master.
- Modbus TCP: Интерфейсът е Ethernet. Данните са в шестнадесетичен формат и се пренасят чрез TCP/IP протокол. Може да има повече от един Master.
- Modbus Plus: Използва специален хардуер. Може да има повече от един Master. Използва се основно от Modicon.

AMK

Control your Motion.

PLControl

Modbus RTU и Modbus TCP:

Modbus RTU:

- Интерфейс RS-232 или RS-485.
- Устройствата се адресират чрез уникален номер.

Modbus TCP:

- Ethernet.
- Modbus RTU + TCP/IP Layer = Modbus TCP.
- Устройствата се адресират чрез IP адрес и номер на устройството.

TCP/IP

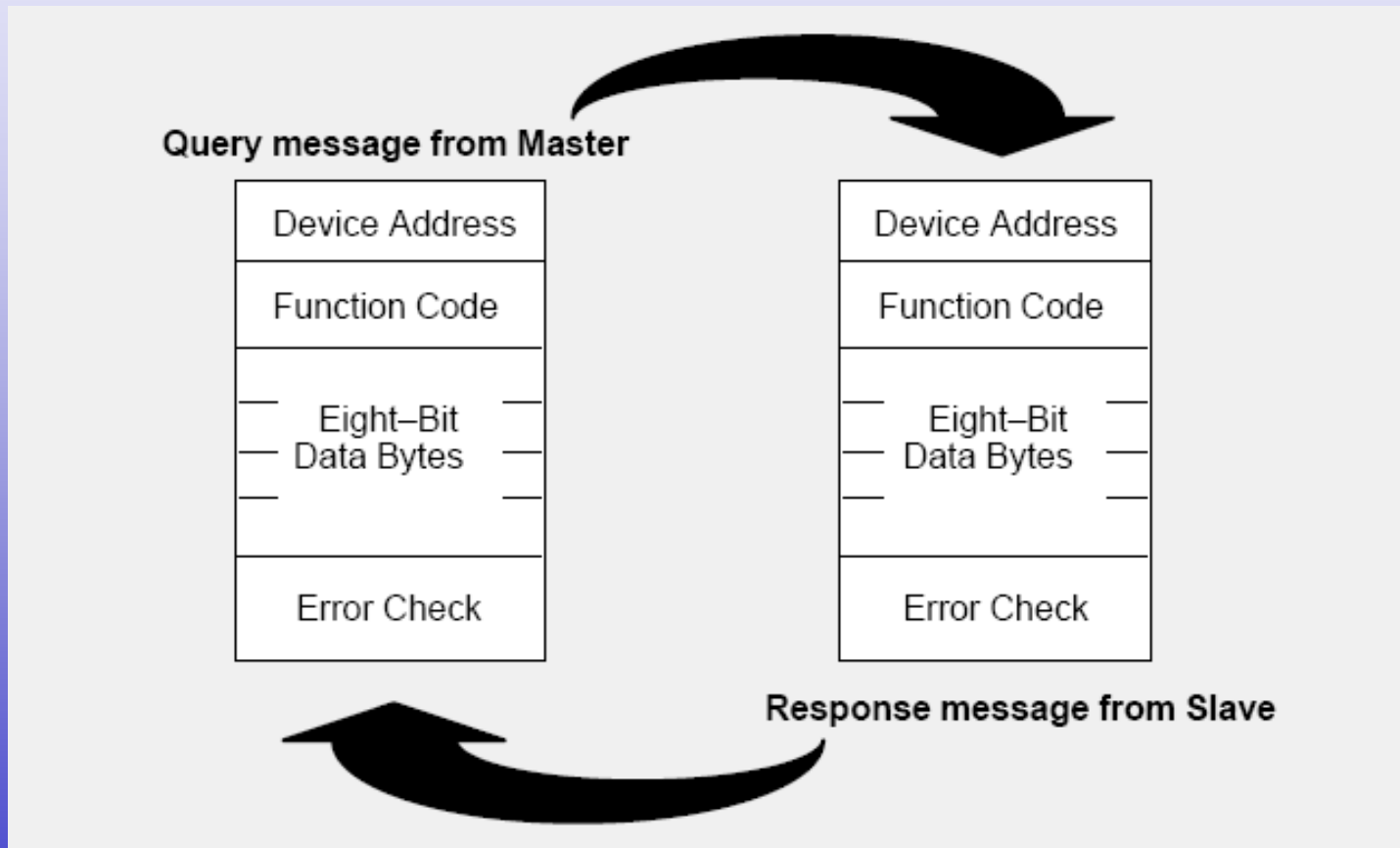
Modbus RTU TCP/IP

TCP/IP

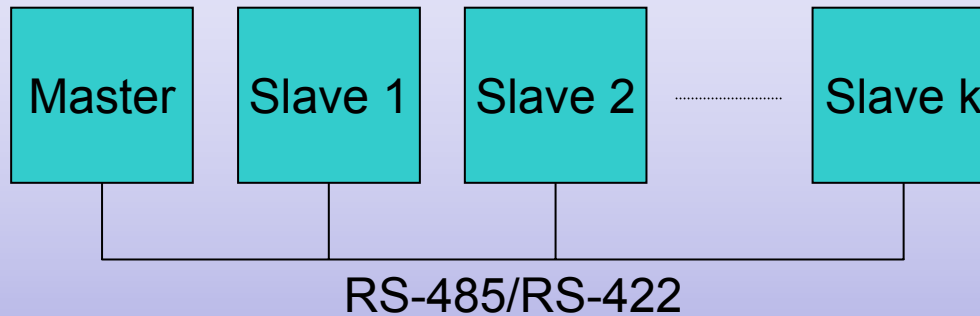
Симеон Трифонов



Modbus – запитване/отговор:



Modbus RTU:



Master/Slave протокол с 1 Master и до 31 Slave-устройства

Формат на телеграмите: номер на устройството (1 байт), код на функцията (1 байт), данни (x байта), CRC16 (2 байта)

Основни функции:

- четене и запис на n думи ($0 < n < 100$);
- четене и запис на n бита ($0 < n < 255$);
- запис на 1 дума;
- запис на 1 бит.

Modbus RTU - телеграми:

Четене на n бита ($0 < n < 255$), FCT = 01H или 02H:

Запитване:

SLAVE	FCT	ADH	ADL	NB OF BITS	CRCH	CRCL
-------	-----	-----	-----	------------	------	------

Отговор:

SLAVE	FCT	NBYTE	...DATA...	CRCH	CRCL
-------	-----	-------	------------	------	------

Четене на n думи ($0 < n < 100$), FCT = 03H или 04H:

Запитване:

SLAVE	FCT	ADH	ADL	NB OF WORD	CRCH	CRCL
-------	-----	-----	-----	------------	------	------

Отговор:

SLAVE	FCT	NBYTE	...DATA...	CRCH	CRCL
-------	-----	-------	------------	------	------



Control your Motion.

PLControl

Modbus RTU - телеграми:

Запис на 1 бит, FCT = 05H:

Запитване:

SLAVE	FCT	ADH	ADL	DATA	DATA	CRCH	CRCL
-------	-----	-----	-----	------	------	------	------

Отговор:

SLAVE	FCT	ADH	ADL	DATA	DATA	CRCH	CRCL
-------	-----	-----	-----	------	------	------	------

Запис на 1 дума, FCT = 06H:

Запитване:

SLAVE	FCT	ADH	ADL	DATA	DATA	CRCH	CRCL
-------	-----	-----	-----	------	------	------	------

Отговор:

SLAVE	FCT	ADH	ADL	DATA	DATA	CRCH	CRCL
-------	-----	-----	-----	------	------	------	------

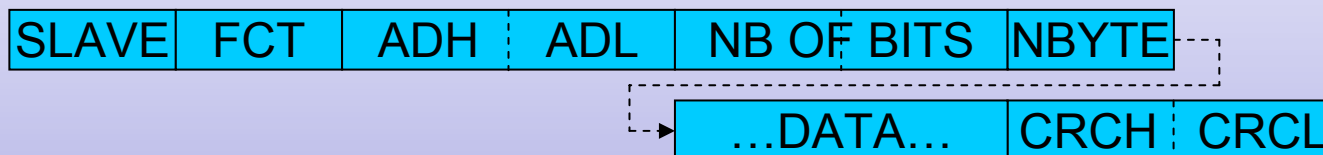
Симеон Трифонов



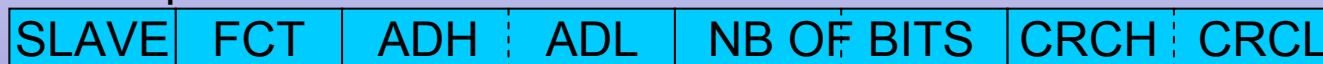
Modbus RTU - телеграми:

Запис на n бита ($0 < n < 255$), FCT = 0FH:

Запитване:

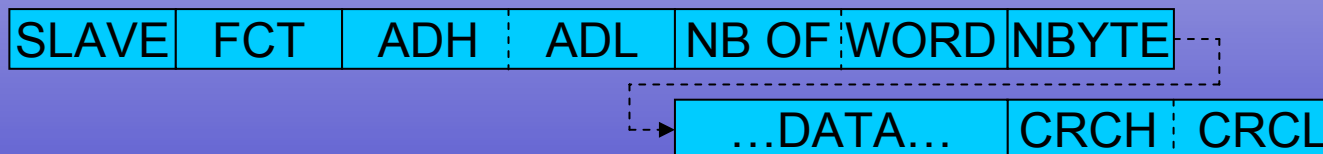


Отговор:



Запис на n думи ($0 < n < 100$), FCT = 10H:

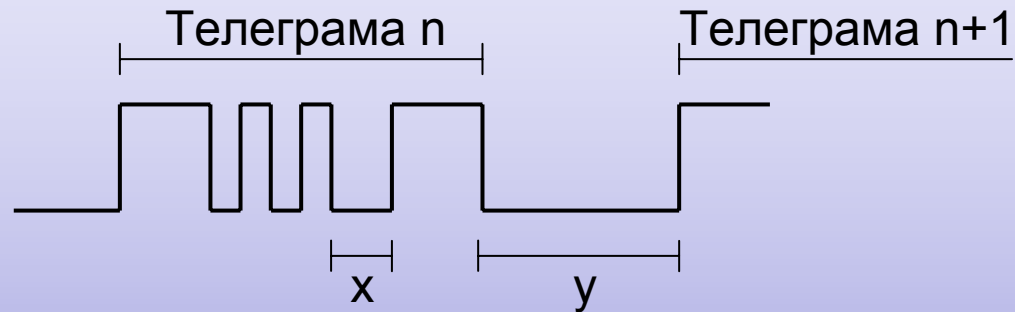
Запитване:



Отговор:



Modbus RTU - тайминг:



Baud rate	Време (ms)
19200	4
9600	5,5
4800	11

Краят на телеграмите се определя по време. Ако за време по-голямо от y по интерфейса не преминават данни, то предходната телеграма е приключена.

AMK

Control your Motion.

PLControl

CAN bus:

CAN bus е промишлен сериен интерфейс, използван за пренос на данни 2 проводника (усукана двойка), подобен на RS-485. Разработен е през 1980г. от Bosch за нуждите на автомобилостроенето. По-късно се стандартизира и получава широко разпространение във всякакъв вид промишлени приложения.

Протоколи от високо ниво:

- DeviceNet;
- CANopen;
- специфични за потребителя протоколи.

Симеон Трифонов



Катедра “Автоматика,
информационна и управляваща
техника”

AMK

Control your Motion.

PLControl

CANopen:

- **Standard Can in Automation CAN 2.0B:**
 - 11Bit-ов идентификатор.
- **CANopen Standard CiA DS 301 Version 4.01:**
 - NMT (Network ManagementT) - контрол на състоянието;
 - Node Guarding - наблюдение на Slave-устройствата;
 - Life Guarding - наблюдение на Master-устройството;
 - Изтрощане / приемане на PDO (Process Data Object) – пренос на специфични за потребителя данни;
 - Client / Server SDO's (Service Data Object) – пренос на служебни данни;
 - Emergency Object - съобщения за грешка;
 - Синхронизационен обект.

Симеон Трифонов



ACC

PLControl

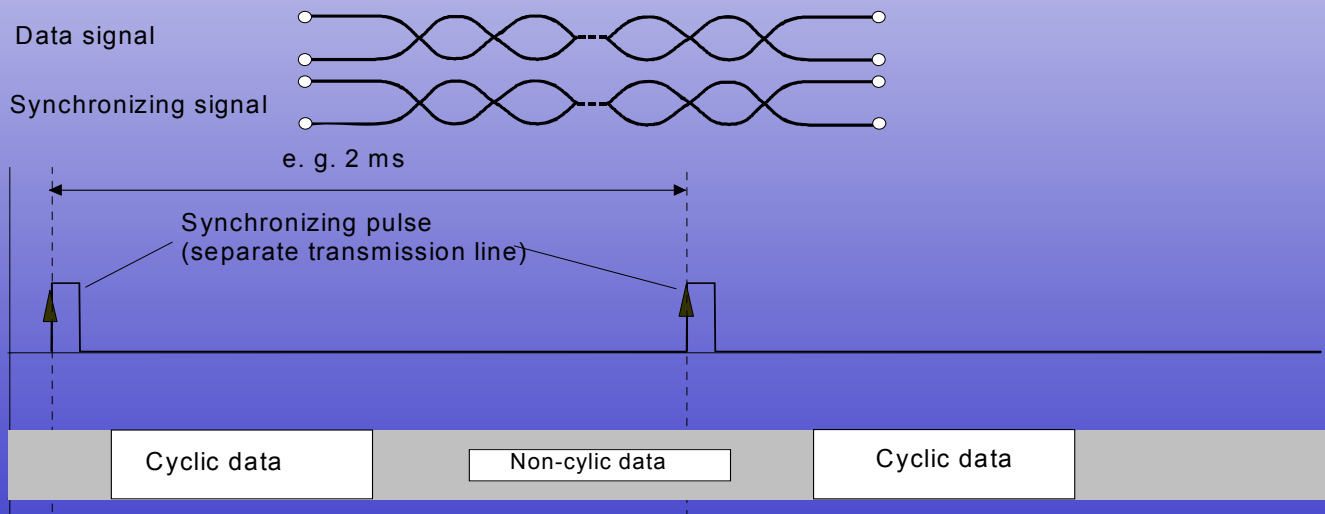
ACC (**A**MK **C**AN **C**ommunication) = CANopen + AMK разширения

Хардуерна синхронизация:

- Slave-устройствата се синхронизират по външен импулс;
- грешка при синхронизацията < 1μs.

Прецизна синхронизация на съобщенията с позиционния контролер.

Симеон Трифонов



PDO (Process Data Object):

- “Real-time” комуникация
- “Multi master” комуникация
- До 8 байта данни на PDO
- SYNC-PDO (синхронно предаване на данни, например зададена позиция, актуална позиция или скорост,...)
- EVENT PDO (асинхронно предаване на данни, например четене и установяване на цифрово входове и изходи)
- “Broadcast” – съобщение (приема се от много устройства едновременно)
- PDO се конфигурират чрез следните параметри:
 - COB-ID (Communication OBject IDentifier) – идентификатор на обекта
 - тип (например синхронни или асинхронни);
 - “mapping” – местонахождение на предаваните/приеманите данни.

PDO "mapping":

Application Interface (API) - област от паметта на серво-задвижването, която дава възможност за достъп до неговите ресурси (diMainSetpoint, diActPosition, byErrSys, wRealTimeBits...)

- **Управление на серво-задвижването директно чрез API-променливи:**

- например diMainSetpoint може да задава момент, скорост или позиция в зависимост от режима на работа на устройството;
- няма нужда от PLC.

- **Управление посредством PLC – обработва данните и управлява серво-задвижването:**

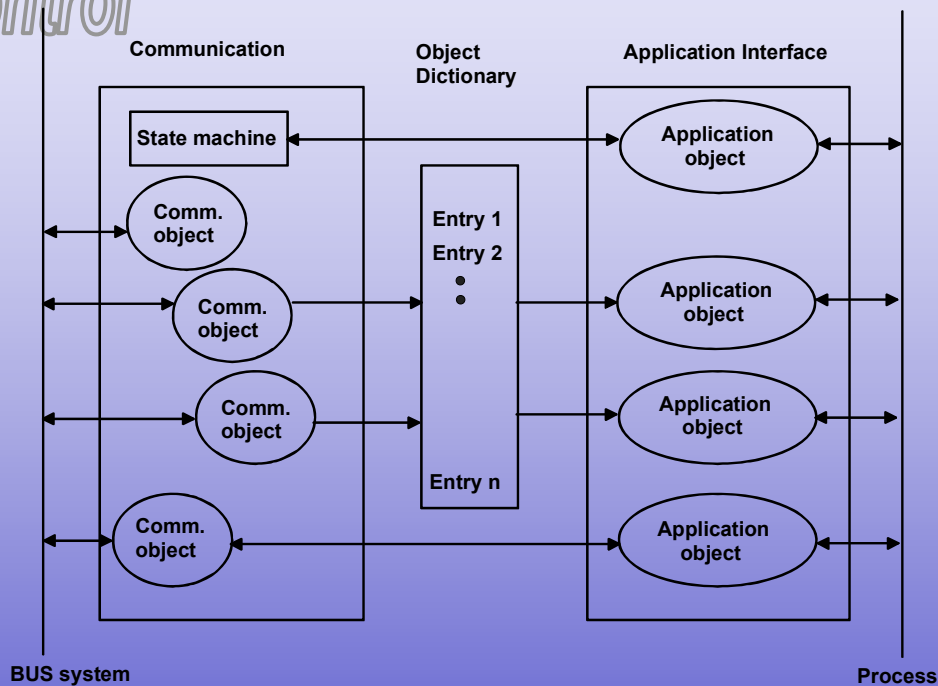
- Long word: lwInX, lwOutX lwSyncInX, lwSyncOutX
- Double word: dwInX, dwOutX dwSyncInX, dwSyncOutX
- Word: wInX, wOutX wSyncInX, wSyncOutX
- Byte: byInX, wOutX bySyncInX, bySyncOutX



SDO (Service Data Object):

- **Асинхронна комуникация**
- **Специфична за AMK SDO функционалност:**
 - Master-устройството има достъп до всички Slave-устройства в мрежата;
 - диагностика (Master-устройството може да прочете съобщенията за грешки на Slave-устройствата);
 - четене и запис на параметрите на устройствата;
 - управление на устройствата;
 - четене на състоянието на устройствата.
- **SDO за четене и запис на параметри в речника на обектите:**
 - специфични за устройството данни (име, тип, версия...);
 - “Guarding” (“guard time”, “life time factor”);
 - Server / Client SDO параметри;
 - конфигурация на PDO и настройка на комуникационни параметри.

CANopen (ACC) – модел на устройствата:



Комуникационни обекти
PDO, SDO, SYNC,
Emergency (EMCY) ,
Network Management (NMT)

Речник с обекти
Интерфейс между
комуникационни
и „application“ - обекти

“Application” - обекти
API, PLC-променливи

Характеристиките на CAN-устройството се описват чрез EDS-файл (**E**lectronic **D**ata **S**heet).



CANopen (ACC) – речника с обекти:

- **Област за комуникационния профил (индекси 1000h - 1FFFh)**

Обекти за обмен на данно между CAN устройствата:

- индекси за PDO и SDO трансфер;
- индекси за Guard Time, Life Time Factor;
- съобщения за грешки...

- **Област за профил, специфичен за производителя (индекси 2000h - 5FFFh)**

Дефиниции на специфични обекти:

- PLC I/O (входни- изходни данни);
- Application Interface (API).

- **Област за стандартен профил (индекси 6000h - 9FFFh)**

- обекти, дефинирани в DS-301.

- **Типове данни (0001h-0009h)**

Речникът съдържа различни профили, които притежават обекти. Всеки обект се адресира чрез 16-битов индекс и 8-битов субиндекс.

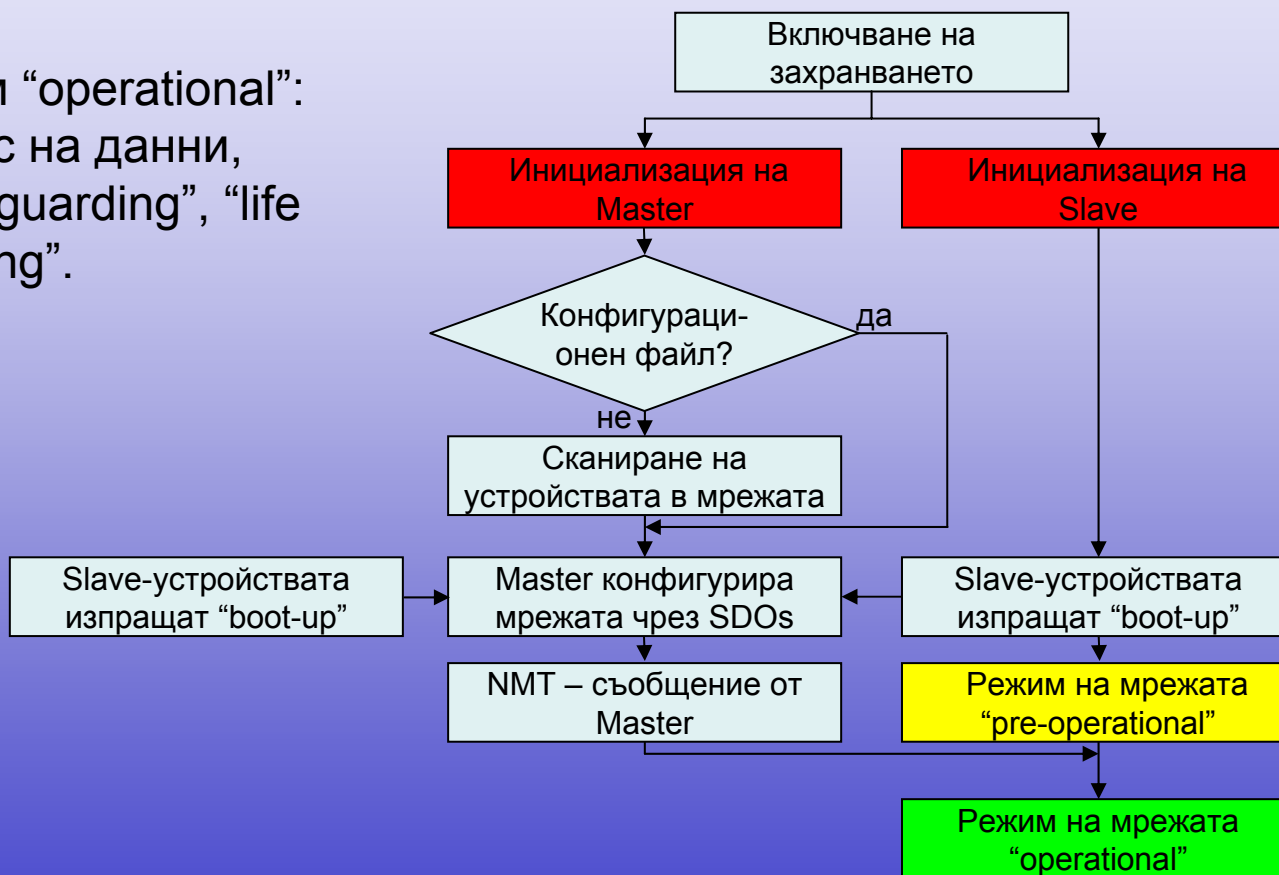
CANopen (ACC) – надеждност на мрежата:

- Node Guarding (специфично за Master-устройствата):
 - Master-устройството циклично изпраща запитване към всяко Slave-устройство;
 - Slave-устройствата трябва да отговорят в рамките на определеното време.
- Life Guarding (специфично за Slave-устройствата):
 - Slave-устройствата чакат запитване от Master-устройството в рамките на определено време.

Ако възникне “guarding” или някакъв вид CAN грешка, CAN Master-устройството получава “Emergency”-съобщение и показва грешката.

CANopen (ACC) – стартиране и режими на работа:

Режим “operational”:
пренос на данни,
“node guarding”, “life
guarding”.



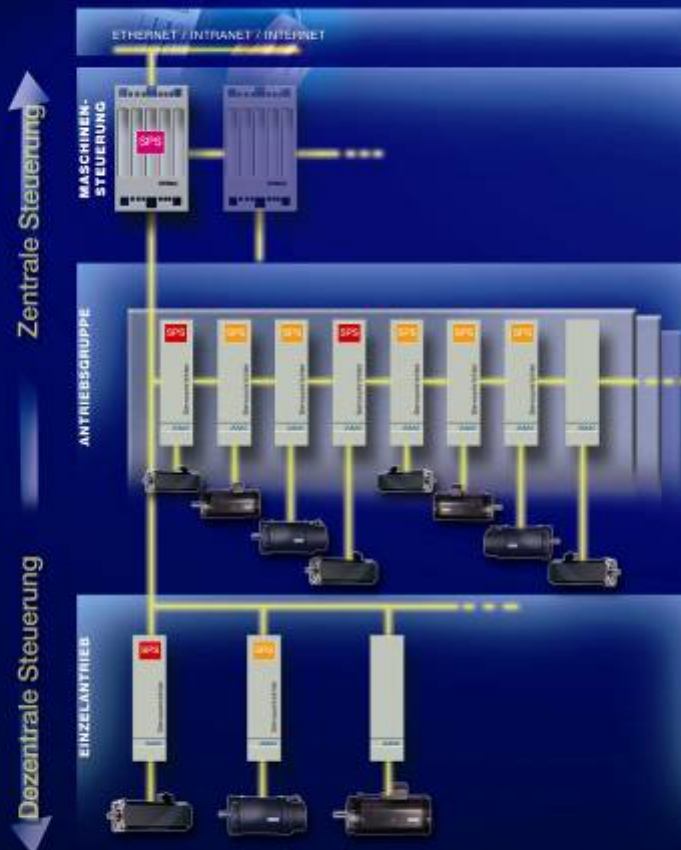
AMK

Control your Motion.

PLControl

Комплексни системи за автоматизация:

Durchgängige SPS- und Motion Control Technologie vom Servoantrieb bis zur Maschinensteuerung.



Zentrale Steuerung

Dezentrale Steuerung

- AMK-SPS-Technologie aus einer Hand nach IEC 61131 mit CoDeSys-Programmierertechnik.
- SPS in PC-basierender AMK-Steuerung
- SPS als leistungsstarke Antriebsoption
- SPS in Standard-Antriebssoftware integriert

AMK
Control your Motion.

Симеон Трифонов



Катедра "Автоматика,
информационна и управляща
техника"

Централизирано управление чрез мощен PLC:



Симеон Трифонов



AMK
Control your Motion.

PLControl

Децентрализирано управление чрез вградени в серво-
задвижванията PLC:



Симеон Трифонов



Катедра "Автоматика,
информационна и управляваща
техника"

AMK

Control your Motion.

PLControl

Унифицирана PLC технология:



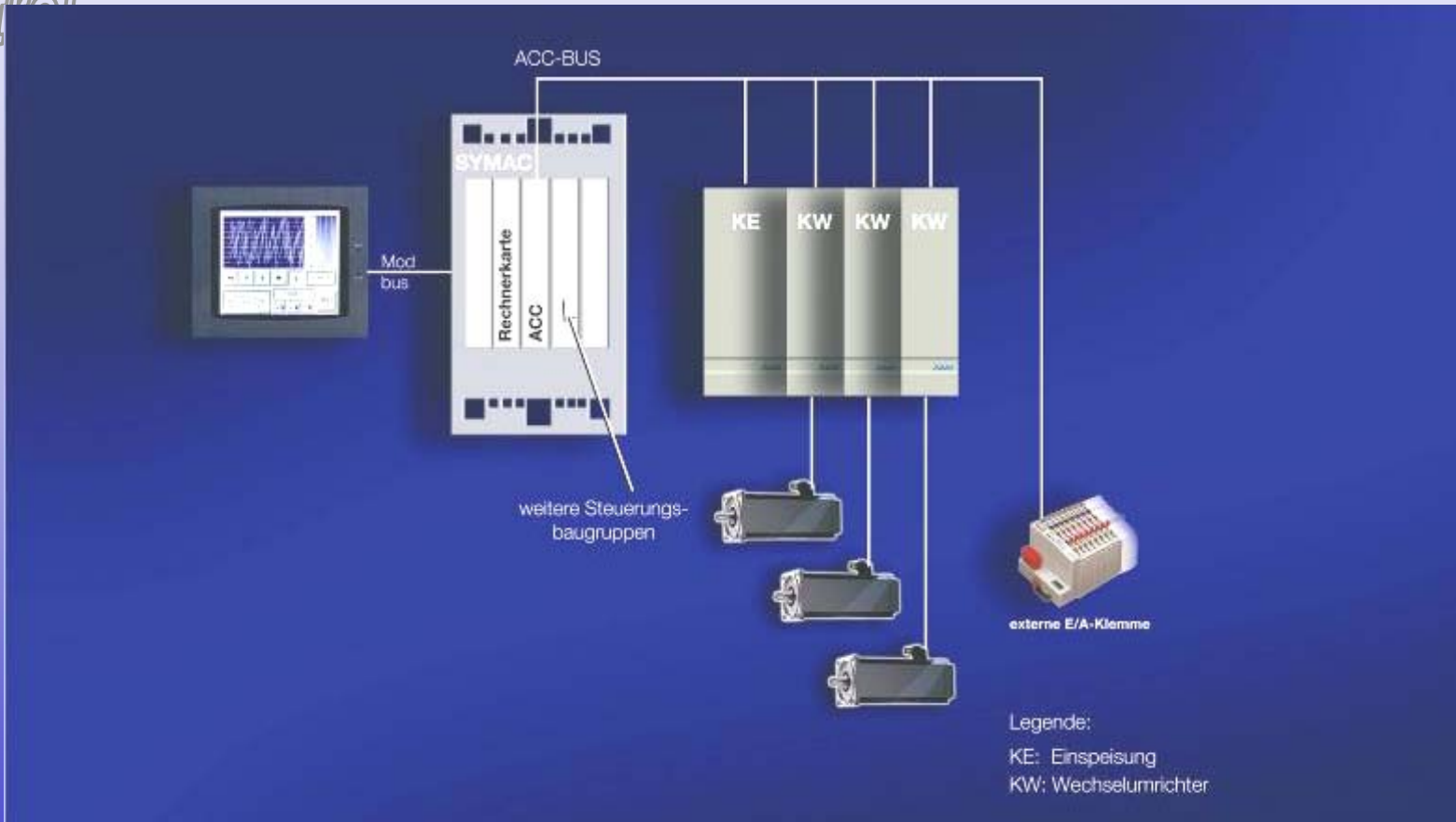
- Унифицирано програмиране и обработка на данните
- Намаляване на инженерния труд и съответно цената
- Програмиране с CoDeSys
- Производителя на PLC предлага специализирани библиотеки за лесно и бързо решаване на специфични проблеми

Симеон Трифонов



Катедра “Автоматика,
информационна и управляваща
техника”

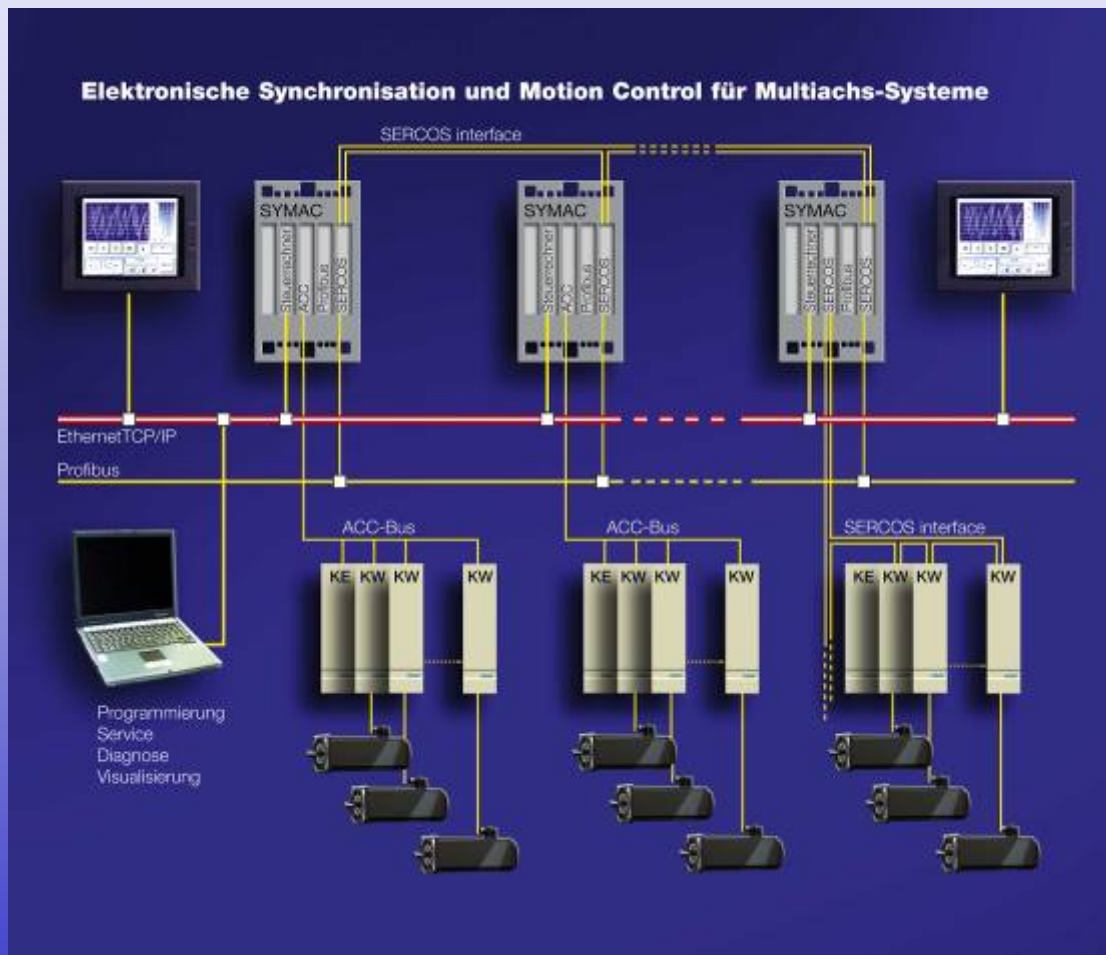
Комуникация в реално време:



Симеон Трифонов



Отворена система:



Симеон Трифонов

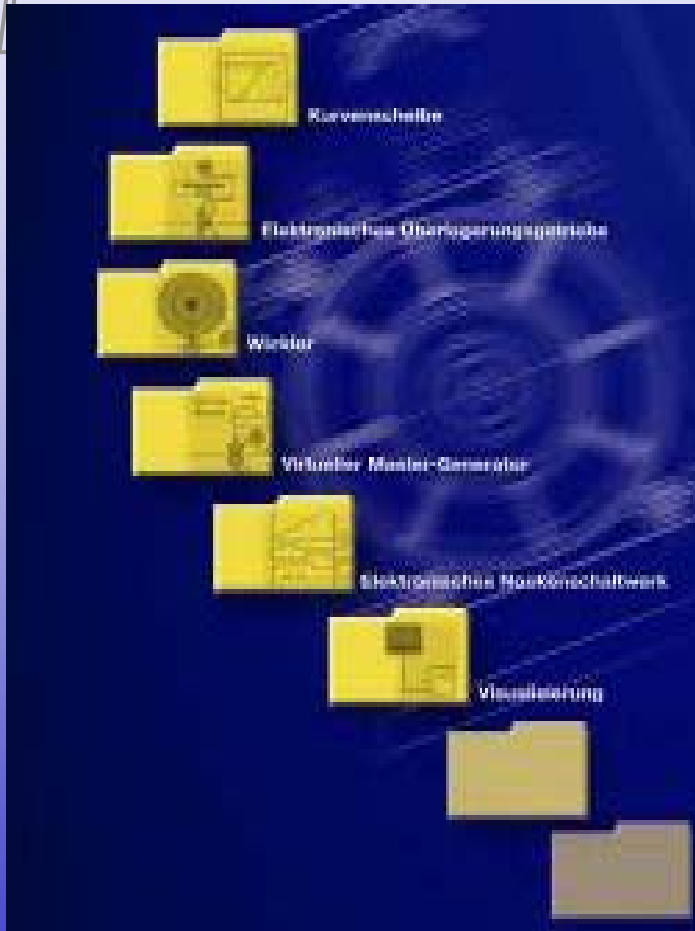


AMK

Control your Motion.

PLControl

Библиотеки, специфични за производителя:



- Позициониране
- Електронна предавка
- Регулатор
- Virtuален Master
- Функционален интерполатор
- Регулиране по печатен маркер
- Електронен САМ-контактор
- Навиване/развиване
- Измервателни функции
- Четене/запис на параметри
- Комуникация

Симеон Трифонов



Катедра "Автоматика,
информационна и управляваща
техника"

AMK

Control your Motion.

PLControl

Какво е CoDeSys:

CoDeSys (**CO**ntroller **DE**velopment **SYS**tem) е среда за разработване на PLC програми на немската фирма 3S GmbH.

Възможности:

- създаване на PLC програми на програмни езици;
- тестване създадените програми;
- “дебъгване” на създадените програми;
- създаване на специфични визуализации и свързване с програмите;
- документиране.



Симеон Трифонов



Катедра “Автоматика,
информационна и управляваща
техника”

AMK

Control your Motion.

PLControl

Какво е IEC61131-3:

IEC61131-3 е интернационален стандарт, който засяга програмирането на PLC.

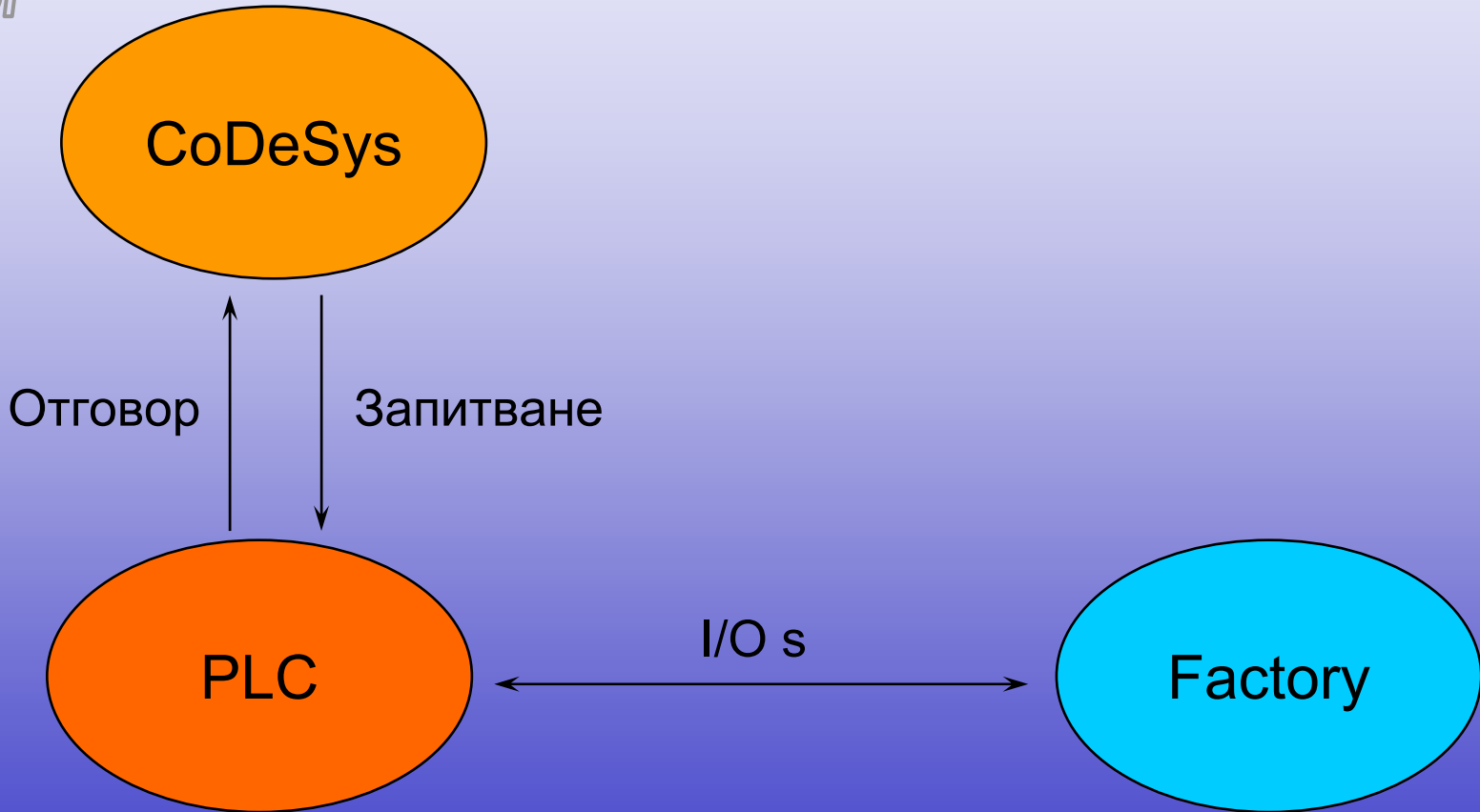
IEC61131-3 дефинира:

- типовете данни;
- структурата на програмите;
- синтаксисът и семантиката на 5 различни езика за програмиране.

Симеон Трифонов



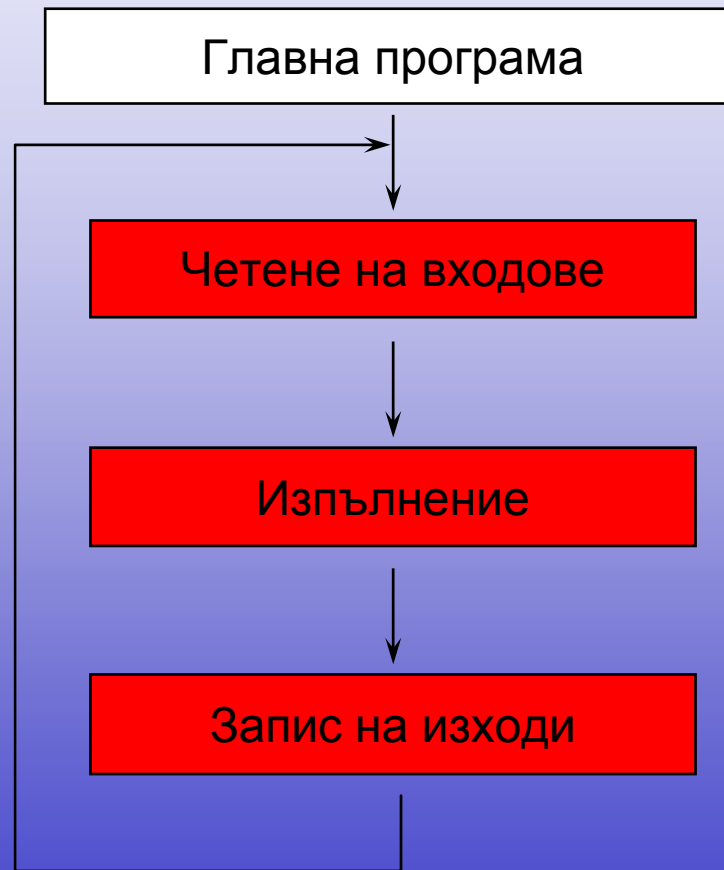
Връзка между CoDeSys и PLC:



Симеон Трифонов



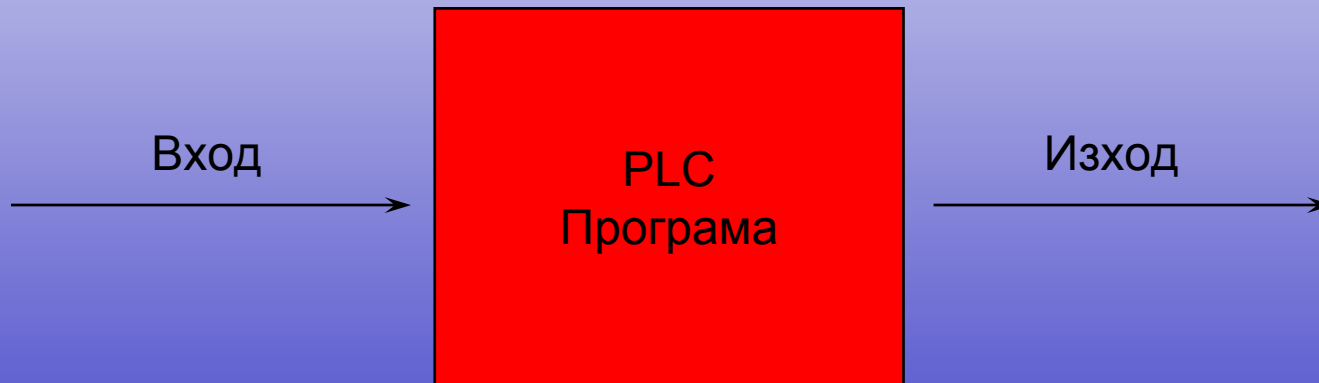
Изпълнение на PLC програмите:



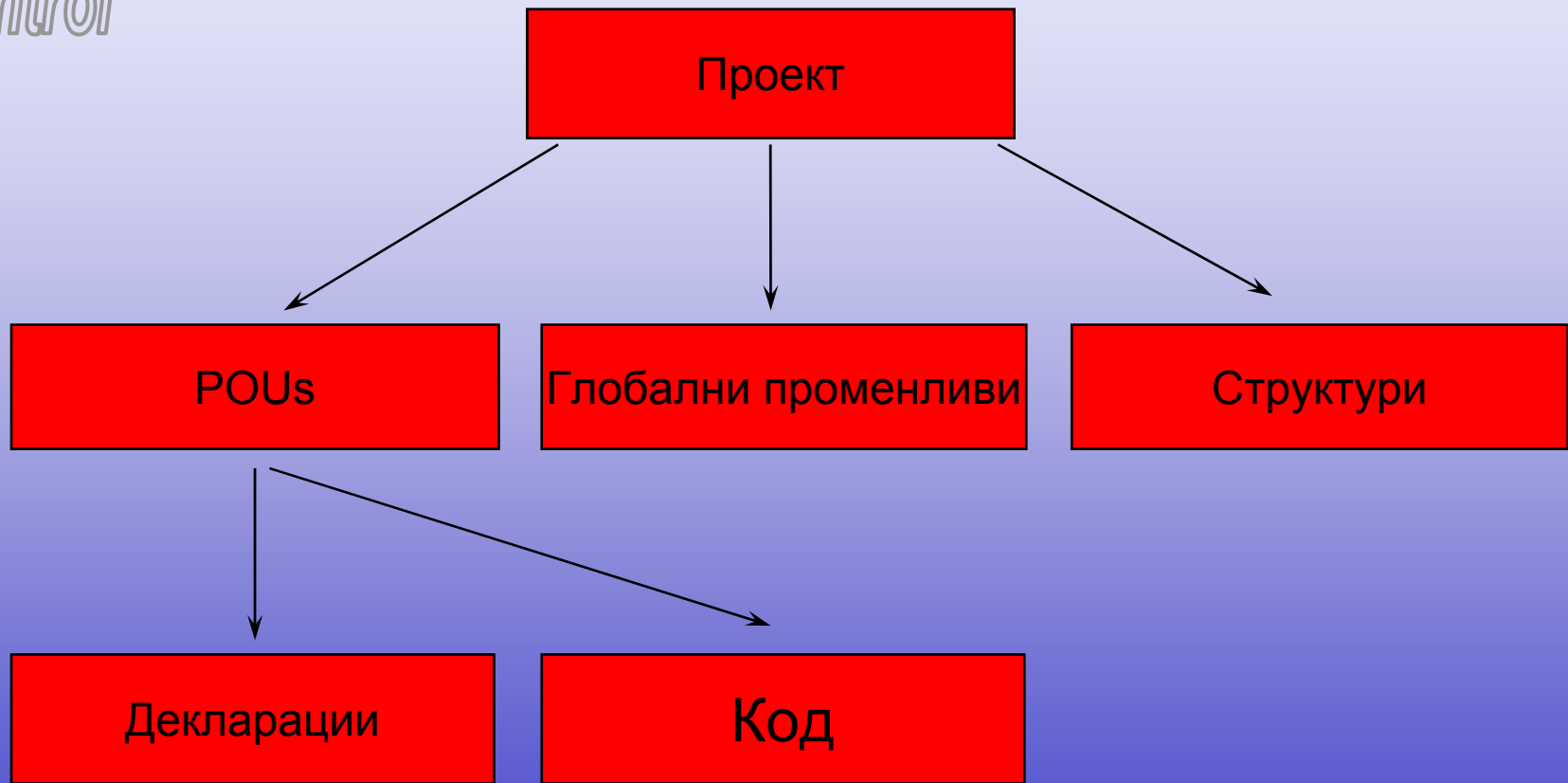
Правила при PLC програмирането:

- Връщане към извикващата функция
- Повишено внимание при използване на цикли (да се държи сметка за времето за изпълнение)
- Повишено внимание при използване на индекси на масиви и указатели

Симеон Трифонов



Структура на PLC програмата:



Симеон Трифонов



Какво е POU?

POU (Program Organization Unit) е програмна единица, съдържаща код, която може да се извиква от друга програмна единица.

В CoDeSys съществува една резервирана програмна единица, наречена **PLC_PRG**, която се извиква от системата (аналогично на main функцията при програмиране на C).

Съществуват три типа POU, които ще бъдат обяснени по-късно.

Данни и адреси:

Валидност:

локални (1 POU) или глобални (всички POU)

CoDeSys поддържа 3 метода на деклариране:

текстово, таблично и автоматично

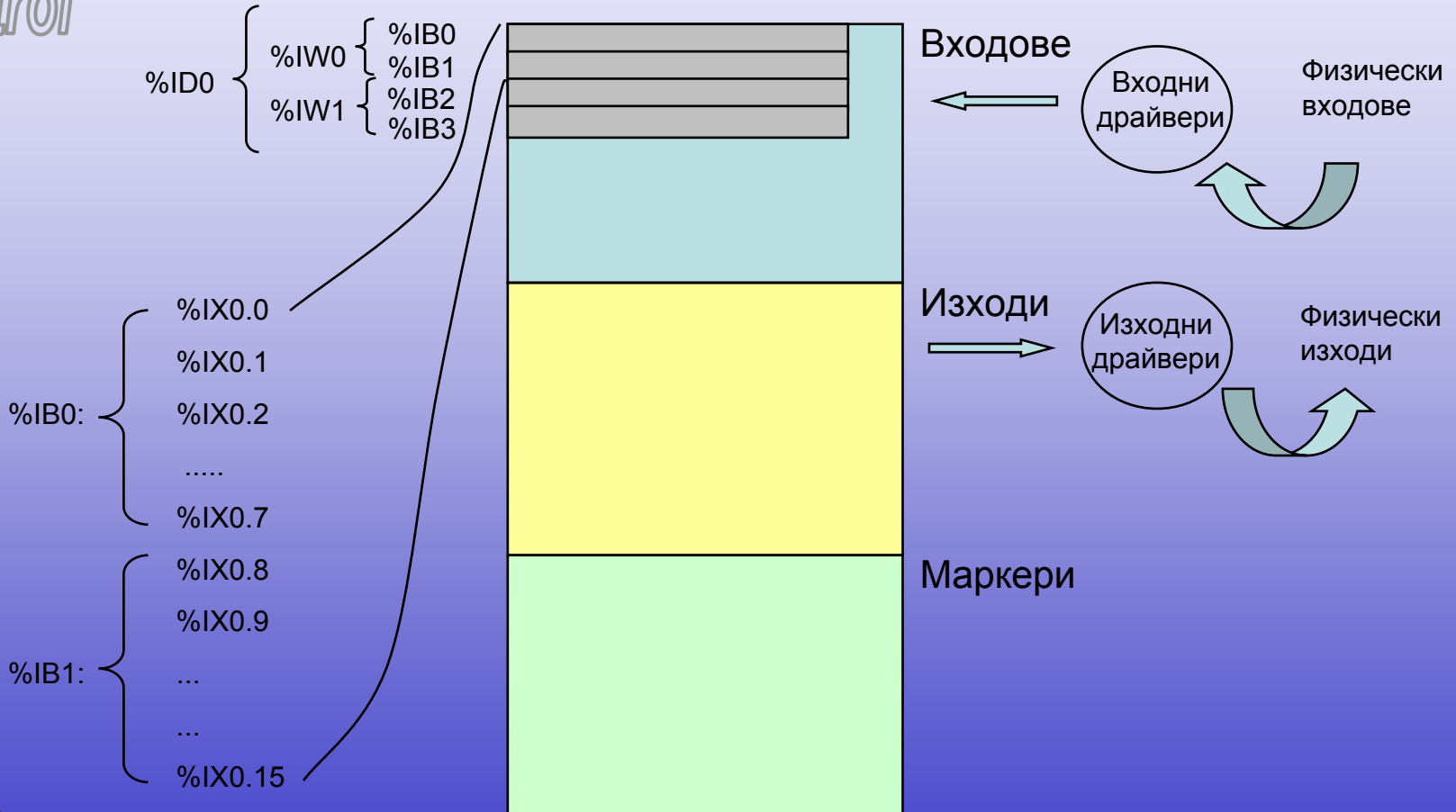
Типове променливи, фиксирани към адреси:

към входна област, към изходна област, към област за маркери

Синтаксис на променливите, фиксирани
към адреси:

- Означават се с ‘%’
- Префикс за областта:
 - I – вход
 - Q – изход
 - M – маркер
- Размер:
 - X – единичен бит
 - B – байт (8 бита)
 - W – дума (16 бита)
 - D – двойна дума (32 бита)
- Примери:
 - %IW215
 - %QX1.1
 - %MD48

“Process image” – изображение на процеса:



Симеон Трифонов



Стандартни типове данни:

Типовете данни, които се използват при създаването на PLC програми са стандартизирани в IEC61131-3:

- BOOL – булева променлива (8 бита, стойности: TRUE, FALSE);
- SINT – **S**hort **I**NTeger (8 бита, стойности: -128 ÷ 127);
- INT – **I**NTeger (16 бита, стойности: -32768 ÷ 32767);
- DINT – **D**ouble **I**NTeger (32 бита, стойности: -2147483648 ÷ 2147483647);
- USINT – **U**nsigned **S**hort **I**NTeger (8 бита, стойност: 0 ÷ 255);
- UINT – **U**nsigned **I**NTeger (16 бита, стойности: 0 ÷ 65535);
- UDINT – **U**nsigned **D**ouble **I**NTeger (32 бита, стойности: 0 ÷ 4294967295);
- BYTE – байт (8 бита, битово адресируема, стойност: 0 ÷ 255);
- WORD – дума (16 бита, битово адресируема, стойности: 0 ÷ 65535);
- DWORD – двойна дума (32 бита, битово адресируема, стойности: 0 ÷ 4294967295);

Стандартни типове данни:

- REAL- числа с плаваща запетая (32 бита, стойност: $1.175494351e-38 \div 3.402823466e+38$);
- LREAL – Long REAL, числа с плаваща запетая (64 бита, стойност: $2.2250738585072014e-308 \div 1.7976931348623158e+308$);
- STRING – текстова променлива (няма ограничение на размера);
Пример: strText: STRING(35):= 'Hello world!';
- TIME – променлива за време (размер и стойности като DWORD);
Пример: tTime: TIME:= t#14ms
tTime1: TIME:= t#12h34m15s
- TOD – Time Of Day (размер и стойности като DWORD);
- DATE – променлива за дата (размер и стойности като DWORD);
Пример: dDate: DATE:= d#1972-03-29;
- DT – Date and Time (размер и стойности като DWORD).

Езици за програмиране, стандартизирани от IEC61131-3:

- ST – Structured Text
- IL – Instruction List.
- FBD – Function Block Diagram
- SFC – Sequential Function Chart
- LD - LaDder

При създаване на PLC програми с CoDeSys, различните програмни единици могат да бъдат написани на различни езици в зависимост от това кой дава най-добри средства за програмиране на съответния алгоритъм.

AMK

Control your Motion.

PLControl

Език за програмиране ST:

- Текстов език
- Език от високо ниво
- PASCAL-ориентиран
- Не много разпространен (това е най-новият език)
- Най-подходящ при условно изпълнение и образуване на цикли (IF, WHILE, FOR, CASE)

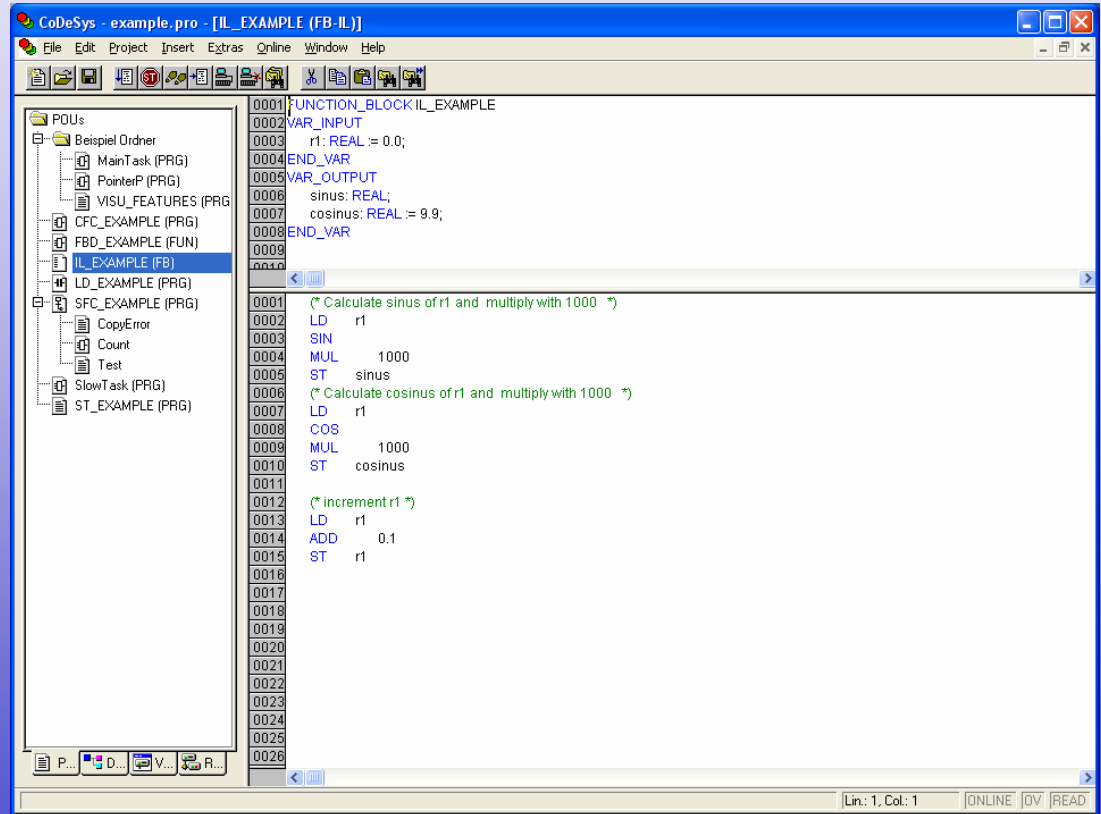
Симеон Трифонов

```
CoDeSys - example.pro - [ST_EXAMPLE (PRG-ST)]
File Edit Project Insert Extras Online Window Help
[Icons]
POUs
  Beispiel Ordner
    MainTask (PRG)
    PointerP (PRG)
    VISU_FEATURES (PRG)
    CFC_EXAMPLE (PRG)
    FBD_EXAMPLE (FUN)
    IL_EXAMPLE (FB)
    LD_EXAMPLE (PRG)
    SFC_EXAMPLE (PRG)
      CopyError
      Count
      Test
      SlowTask (PRG)
      ST_EXAMPLE (PRG)
0001 PROGRAM ST_EXAMPLE
0002 VAR
0003   xVal:INT:= 0;
0004   yVal:INT:= -250;
0005   bottom: INT:= -250;
0006   stripes: ARRAY[0..5] OF INT := 0, 90, 180, 270, 360, 450;
0007   stelle: INT;
0008   i: INT;
0009   rot: INT;
0010   y: BOOL;
0011   wait: INT:= 0;
0012   inv: BOOL:= TRUE;
0013   invZ: BOOL:= FALSE;
0014   scale: INT:= 1000;
0015   run:BOOL:=FALSE;
0016   run_string:STRING(20):='Start';
0017   offset: INT:= 2;
0018 END_VAR
0001 run_string:='Start';
0002 IF NOT run THEN
0003   RETURN;
0004 END_IF;
0005 run_string:='Stop';
0006
0007 rot := rot + offset;
0008
0009 IF (yVal < 0) THEN
0010   yVal := yVal + offset;
0011   bottom := yVal + offset;
0012 ELSE
0013   IF (xVal < 470) THEN
0014     xVal := xVal+offset;
0015   END_IF
0016   IF (bottom > -250) THEN
0017     bottom := bottom -offset;
0018   END_IF
```



Език за програмиране IL:

- Текстов език
- Подобен на асемблер
- Най-добре познатият език в Европа
- Всички оператори работят със специален регистър, наречен акумулатор (LD, ST)
- Удобен при малки програми



The screenshot shows the CoDeSys software interface for editing an IL program. The window title is "CoDeSys - example.pro - [IL_EXAMPLE (FB-IL)]". The left sidebar shows a project tree with "IL_EXAMPLE (FB)" selected. The main editor displays the following IL code:

```
0001 FUNCTION_BLOCK_IL_EXAMPLE
0002 VAR_INPUT
0003   r1: REAL := 0.0;
0004 END_VAR
0005 VAR_OUTPUT
0006   sinus: REAL;
0007   cosinus: REAL := 9.9;
0008 END_VAR
0009
0010
0011
0012 (* Calculate sinus of r1 and multiply with 1000 *)
0013 LD   r1
0014 SIN
0015 MUL   1000
0016 ST   sinus
0017 (* Calculate cosinus of r1 and multiply with 1000 *)
0018 LD   r1
0019 COS
0020 MUL   1000
0021 ST   cosinus
0022
0023 (* increment r1 *)
0024 LD   r1
0025 ADD   0.1
0026 ST   r1
```

Език за програмиране FBD:

- Графичен език
- Удобен, когато няма цикли и много разклонения в програмата
- Състои се от блокове и оператори
- Много разпространен в последно време

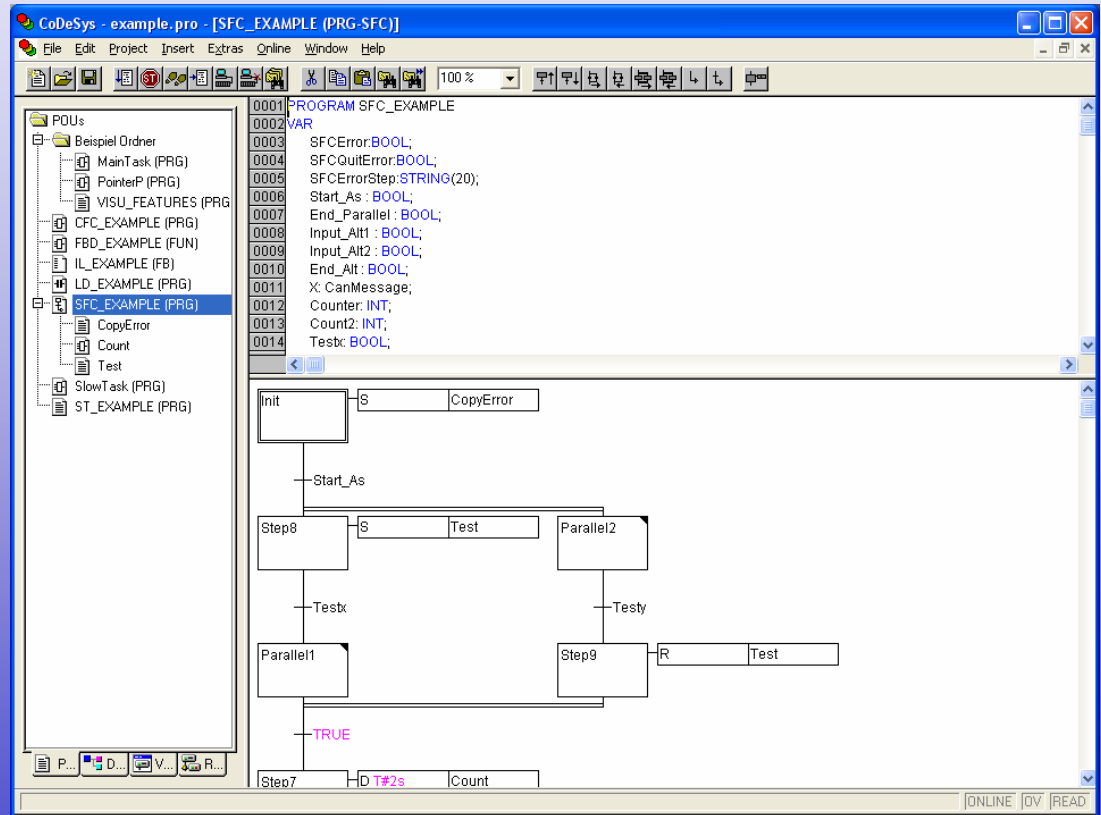
The screenshot displays the CoDeSys software environment. The title bar reads 'CoDeSys - example.pro - [FBD_EXAMPLE (FUN-FBD)]'. The menu bar includes 'File', 'Edit', 'Project', 'Insert', 'Extras', 'Online', 'Window', and 'Help'. The toolbar contains various icons for file operations and execution. The main workspace is divided into three sections:

- Project Tree (Left):** Shows a hierarchy of files including 'Beispiel Ordner', 'MainTask (PRG)', 'PointerP (PRG)', 'VISU_FEATURES (PRG)', 'CFC_EXAMPLE (PRG)', 'FBD_EXAMPLE (FUN)', 'IL_EXAMPLE (FB)', 'LD_EXAMPLE (PRG)', 'SFC_EXAMPLE (PRG)', 'CopyError', 'Count', 'Test', 'SlowTask (PRG)', and 'ST_EXAMPLE (PRG)'. 'FBD_EXAMPLE (FUN)' is selected.
- Code Editor (Top):** Contains the following text:

```
0001 FUNCTION FBD_EXAMPLE:BOOL
0002
0003 .....
0004 Example for a function written in FBD
0005 .....
0006 VAR_INPUT
0007   VAR1:BYTE;
0008   VAR2:BYTE;
0009   VAR3:BYTE;
0010 END_VAR
0011 VAR
0012   b1:BOOL;
0013   b2:BOOL;
0014   b3:BOOL;
0015   b4:BOOL;
0016   b5:BOOL;
0017   b6:BOOL;
0018   b7:BOOL;
0019   b8:BOOL;
```
- Diagram (Bottom):** A graphical ladder logic diagram labeled 'Label1:'. It shows the following logic:
 - Var1 (10) is connected to a MOD block.
 - Var2 (20) is connected to a MOD block.
 - The output of the first MOD block goes to an LT block with a value of 5.
 - The output of the second MOD block goes to a GT block with a value of 10.
 - The outputs of the LT and GT blocks are connected to an AND block.
 - Var2 (20) is also connected to a second MOD block.
 - The output of the second MOD block goes to a GT block with a value of 10.
 - The outputs of the AND block and the second GT block are connected to an OR block.
 - The output of the OR block is connected to the function result output 'FBD_EXAMPLE'.

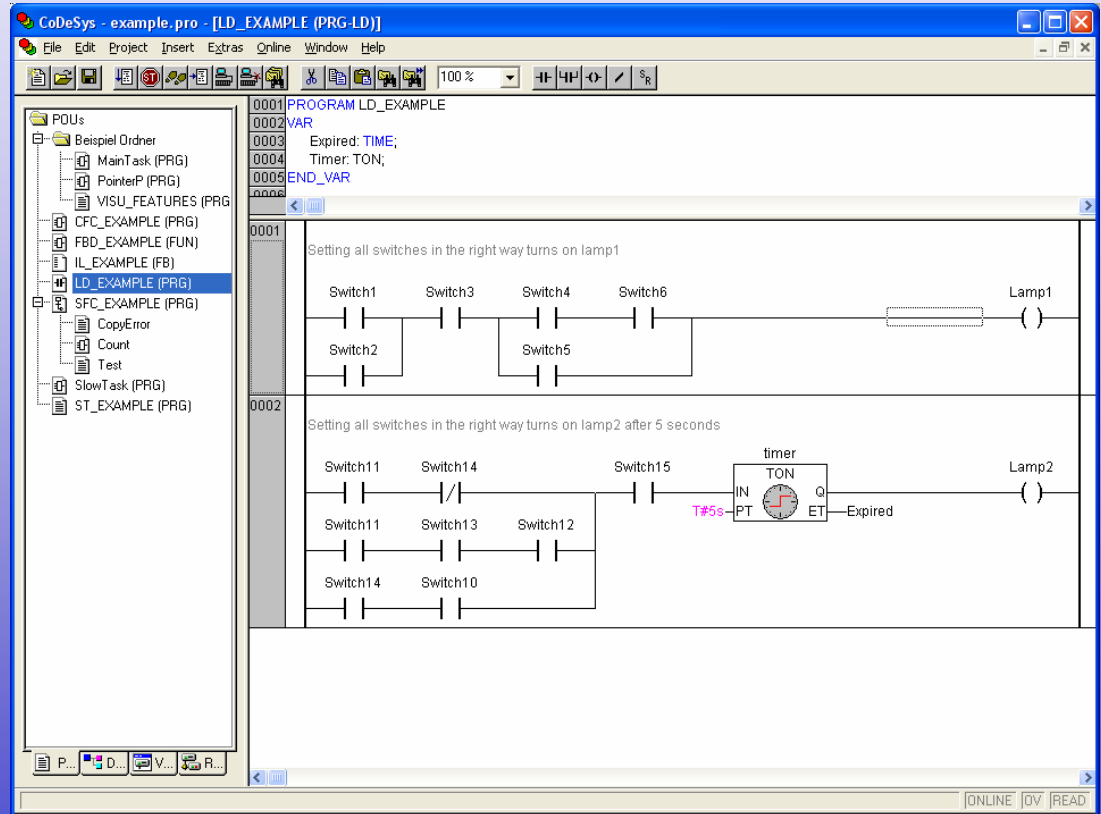
Език за програмиране SFC:

- Графичен език
- Добро структуриране на програмата
- Състои се от стъпки и преходи
- Стъпките съдържат програми на всеки стандартен език
- Не може да се замени с друг език (няма аналог)



Език за програмиране LD:

- Графичен език
- Поддържан от почти всички PLC
- Удобен при програмиране на булева логика
- Много използван в USA, GB и Азия
- Почти неизползваем при работа с аналогови величини



Операции в CoDeSys:

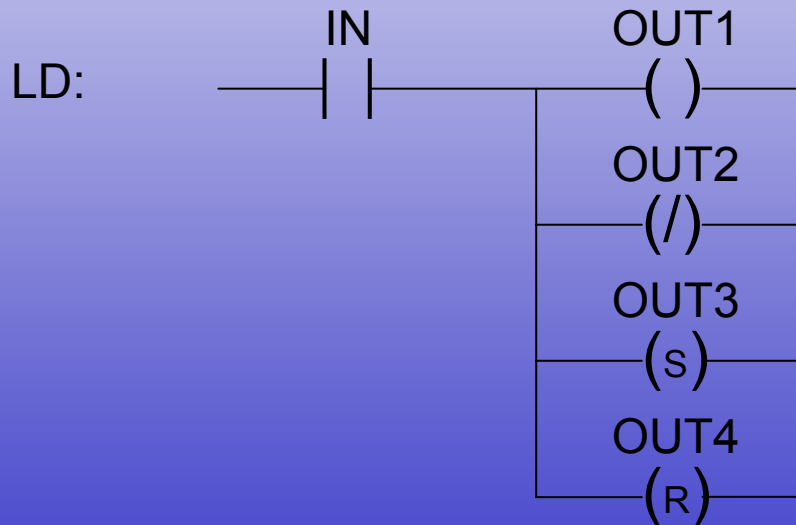
- присвояване;
- булеви операции;
- аналогови операции;
- сравняване;
- селектиране;
- преобразуване на тип;
- операции с реални числа;
- побитово преместване;
- специални операции.

Операция присвояване:

IL: LD, LDN, ST, STN, S, R

ST: A := B;

FBD: A ————— B
A ————— **S** C
A ○ ————— D



Булеви операции:

AND, OR, XOR и NOT

Операциите AND, OR и XOR могат да се изпълняват върху неограничен брой входове.

Когато се прилагат върху данни от тип BOOL, резултатът е TRUE или FALSE.

Когато се прилагат върху данни от тип BYTE, WORD, DWORD, резултатът се получава след побитово изпълнение на съответната операция.

Пример на IL:

Var1: BYTE;

LD 2#10010011

AND 2#10001010

ST Var1 (* Result is 2#10000010 *)

Аналогови операции:

<u>IL,FBD,LD</u>	<u>ST</u>
ADD	+
SUB	-
MUL	*
DIV	/
MOD	MOD

Тези операции могат да се извършват с всеки тип данни, различен от булев.

Сравняване:

<u>IL, FBD, LD</u>	<u>ST</u>
EQ	=
NE	<>
GE	>=
GT	>
LE	<=
LT	<

Тези операции могат да се извършват с всеки тип данни,
различен от булев.

Селектиране:

MIN – намиране на най-малката стойност

MAX – намиране на най-голяма стойност

Пример: `Var1:=MAX(30,40); (* Result is 40 *)`

SEL – избор между две стойности в зависимост от
съдържанието на булева променлива

Пример: `Var1:=SEL(TRUE,3,4); (* Result is 4 *)`

MUX – мултиплексор

Пример: `Var1:=MUX(0,30,40,50,60,70,80); (* Result is 30 *)`

LIMIT – ограничаване

Тези операции могат да се извършват с всеки тип данни,
различен от булев.

Преобразуване на тип:

Формат:

function<type>_TO_<type>

Пример: Var1:= DINT_TO_INT(Var2);

Преобразуването на типове може да стане по подразбиране в случай ако:
 $\text{typewidth}(A) \geq \text{typewidth}(B)$

Внимание:

1. Функциите изискват точния тип данни както в техните декларации.
2. Преобразуването на типове трябва да се избягва когато е възможно.

AMK

Control your Motion.

PLControl

Операции с реални числа:

ABS – абсолютна стойност

SQRT – корен квадратен

LN – натурален логаритъм

LOG – десетичен логаритъм

EXP – експоненциална функция (e^x)

SIN, COS, TAN, ASIN, ACOS, ATAN – тригонометрични функции

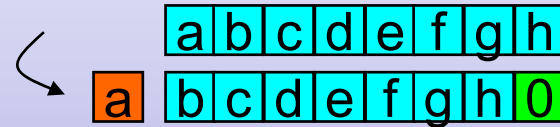
EXPT – експоненциална функция на променлива спрямо друга променлива (X^Y)

Симеон Трифонов

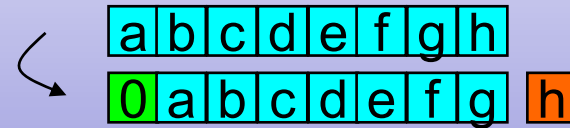


Побитово преместване:

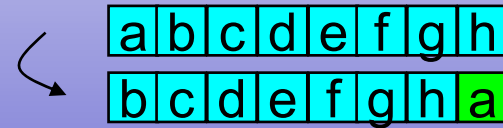
SHL (SHift Left)



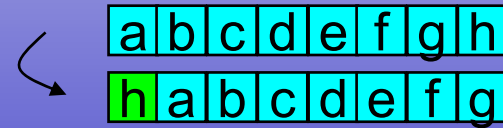
SHR (SHift Right)



ROL (ROtate Left)



ROR (ROtate Right)



Специални операции:

ADR – връща адреса на променливата

Пример: Var1: WORD:= 5;
Ptr: POINTER TO WORD;

Ptr:= ADR(Var1); (*Result is 16#02CC4040 *)

^ - взема съдържанието на паметта, сочена от променливата

Пример: Var1: WORD;
Ptr: POINTER TO WORD:= 16#02CC4040;

Var1:= Ptr^; (* Result is 5);

Типове POU:

Функция (Function):

Приема входни променливи, връща резултат в една променлива, не заема памет.

Функционален блок (Function block):

Използва входни и изходни променливи, заема памет. Могат да се дефинират няколко копия (инстанции) на един функционален блок.

Програма (Program):

Глобално дефиниран функционален блок. Не може да има копия (инстанции).



AMK

Control your Motion.

PLControl

Функции:

- Не могат да запазват използвани данни
- Локалните променливи се инициализират преди всяко изпълнение
- Името на функцията е променливата, която се използва за връщане на резултат, т.е. функцията има определен тип
- Подходящи за извършване на сложни изчисления
- Използването на глобални променливи е забранено

Симеон Трифонов



AMK

Control your Motion.

PLControl

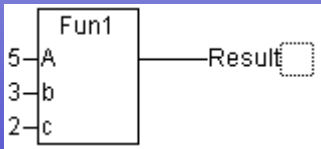
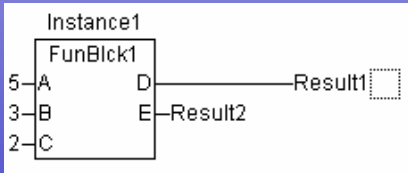
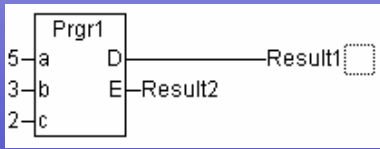
Функционални блокове:

- Всички използвани променливи запазват своите стойности
- Чрез инстанциите се създават копия на набора от променливи, използвани във функционалния блок
- Подходящи за създаване на програмни елементи като броячи, таймери, тригери или други подобни, които изискват запомняне на състоянията на използваните променливи

Симеон Трифонов



Извикване на различните видове POU:

	Function	Function Block	Program
Example	Function Fun1:INT 3 Inputs (INT): A, B, C	Function_Block FunBlck1 3 Inputs (INT): A, B, C 2 Outputs (INT): D, E Instance: Instance1	Program Prgr1 3 Inputs (INT): A, B, C 2 Outputs (INT): D, E
IL	LD 5 Fun1 3,2 ST Result	CAL Instance1(A:=5, B:=3, C:=2) ... LD Instance1.D ST Result1 LD Instance1.E ST Result2	CAL Prgr1(a := 5, b := 3, c := 2) ... LD Prgr1.D ST Result1 LD Prgr1.E ST Result2
ST	Result:=Fun1(5,3,2);	Instance1(A:=5, B:=3, C:=2); ... Result1:=Instance1.D; Result2:=Instance1.E;	Prgr1(a := 5, b := 3, c := 2); ... Result1:= Prgr1.D; Result1:= Prgr1.E;
FBD / LD			

Симеон Трифонов



AMK

Control your Motion.

PLControl

Библиотеки с готови POU:

- Библиотеките съдържат готови обекти, които могат да се използват в различни проекти
- Могат лесно да се създават собствени библиотеки със средствата на CoDeSys
- Производителите на PLC могат да разпространяват готови криптирани библиотеки (защитено “Know How”)
- CoDeSys позволява да се извикват функции, написани на C като библиотечни елементи
- CoDeSys предоставя за използване стандартна библиотека (standard.lib) с често използвани обекти – за работа със стрингове, за регистриране на активен фронт, броячи, таймери и т.н.

Симеон Трифонов



Функции за работа със стрингове:

LEN – определя размера на стринга

LEFT – връща определен брой символи от ляво на дясно на подадения като параметър стринг

RIGHT - връща определен брой символи от дясно на ляво на подадения като параметър стринг

MID – връща определен брой символи от определена позиция на подадения като параметър стринг

CONCAT – слепва два стринга

INSERT – вмъква един стринг от определена позиция на друг стринг

DELETE – премахва определен брой символи от определена позиция на подадения като параметър стринг

REPLACE – подменя определен брой символи от определена позиция

FIND – търси един стринг в друг и връща позицията, от която първия стринг започва

Регистриране на активен фронт:

R_TRIG – регистрира преден фронт (FALSE -> TRUE)

Пример: RTRIGInst(CLK:= VarBOOL1);
VarBOOL2 := RTRIGInst.Q;

F_TRIG – регистрира заден фронт (TRUE -> FALSE)

Пример: CAL FTRIGInst(CLK := VarBOOL1)
LD FTRIGInst.Q
ST VarBOOL2

Броячи:

CTU – броене нагоре (увеличава стойността си при всеки активен фронт на броячния вход)

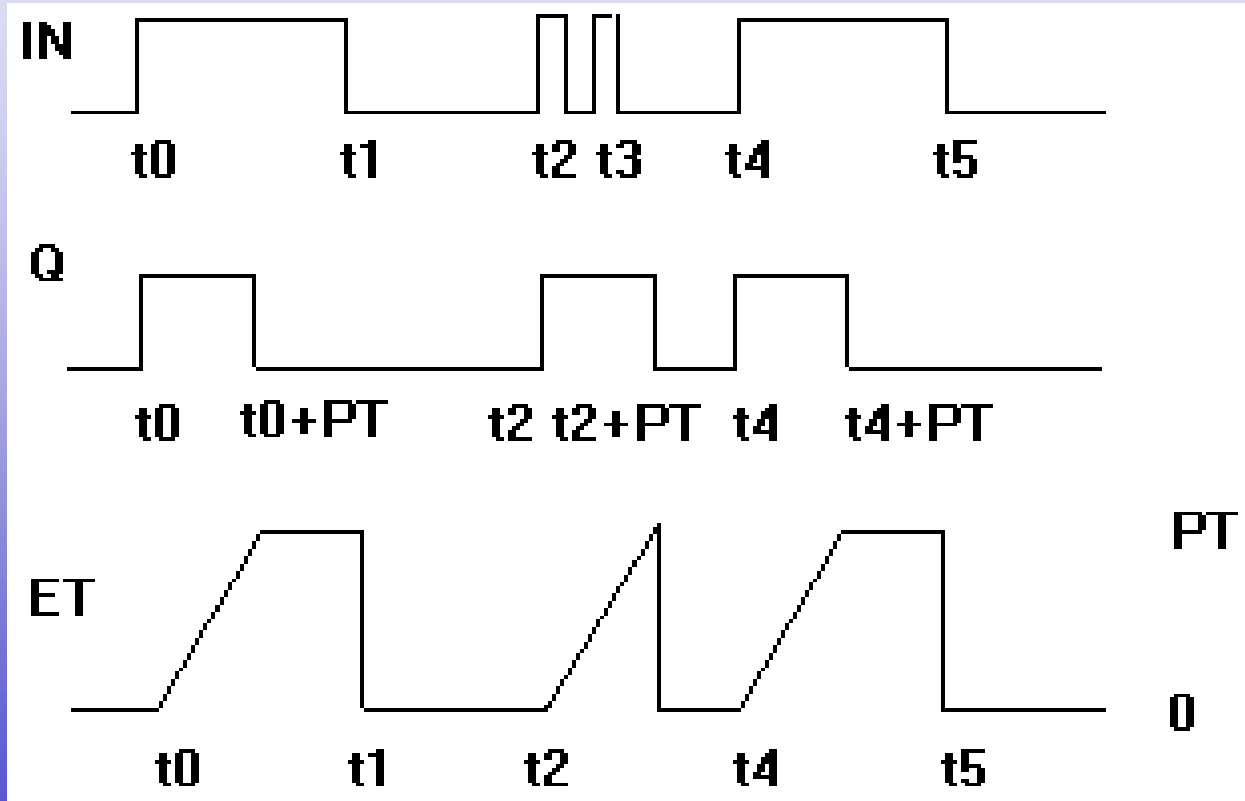
Пример: CTUInst(CU:= VarBOOL1, RESET:=VarBOOL2 ,
 PV:= VarINT1);
 VarBOOL3 := CTUInst.Q ;
 VarINT2 := CTUInst.CV;

CTD - броене надолу (намалява стойността си при всеки активен фронт на броячния вход)

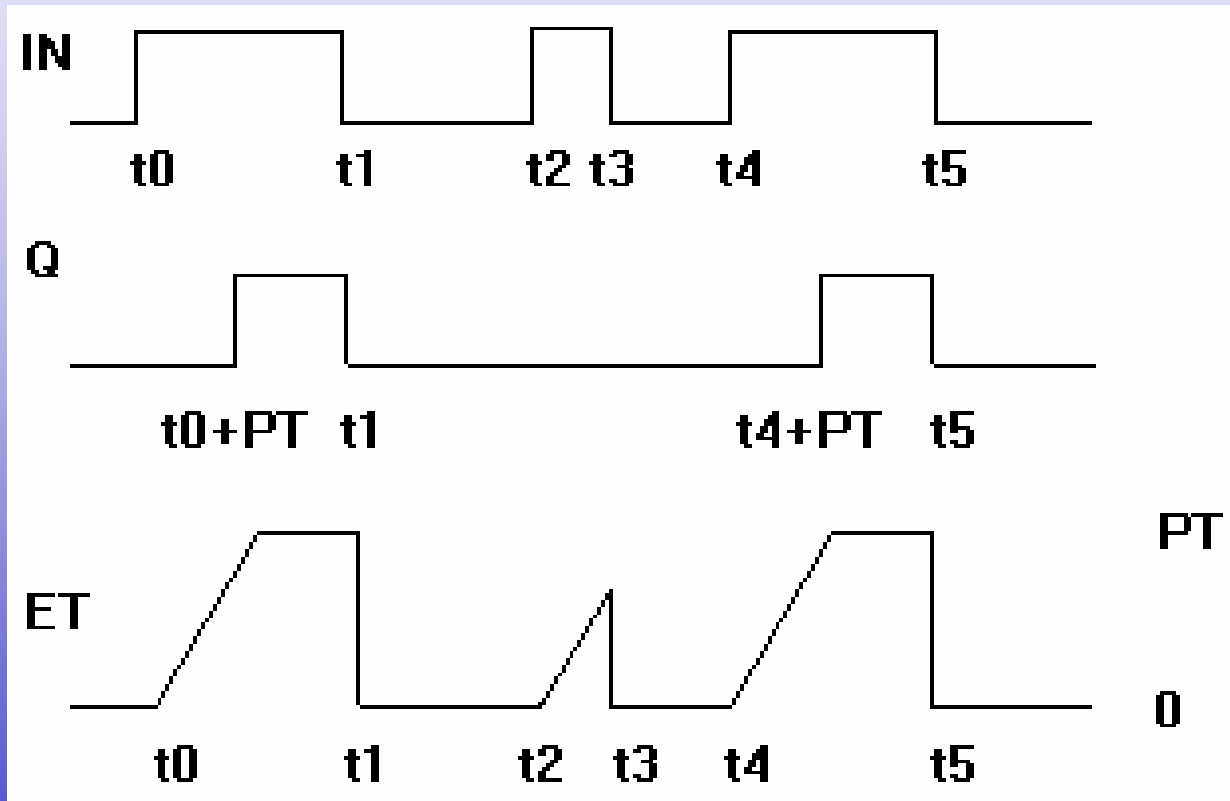
CTUD – броене нагоре или надолу (увеличава или намалява стойността си при всеки активен фронт на един от двата входа)

Пример: CTUDInst(CU := VarBOOL1, CU:= VarBOOL2,
 RESET := VarBOOL3, LOAD:=VarBOOL4 , PV:= VarINT1);
 VarBOOL5 := CTUDInst.QU ;
 VarBOOL6 := CTUDInst.QD ;
 VarINT2 := CTUDInst.CV;

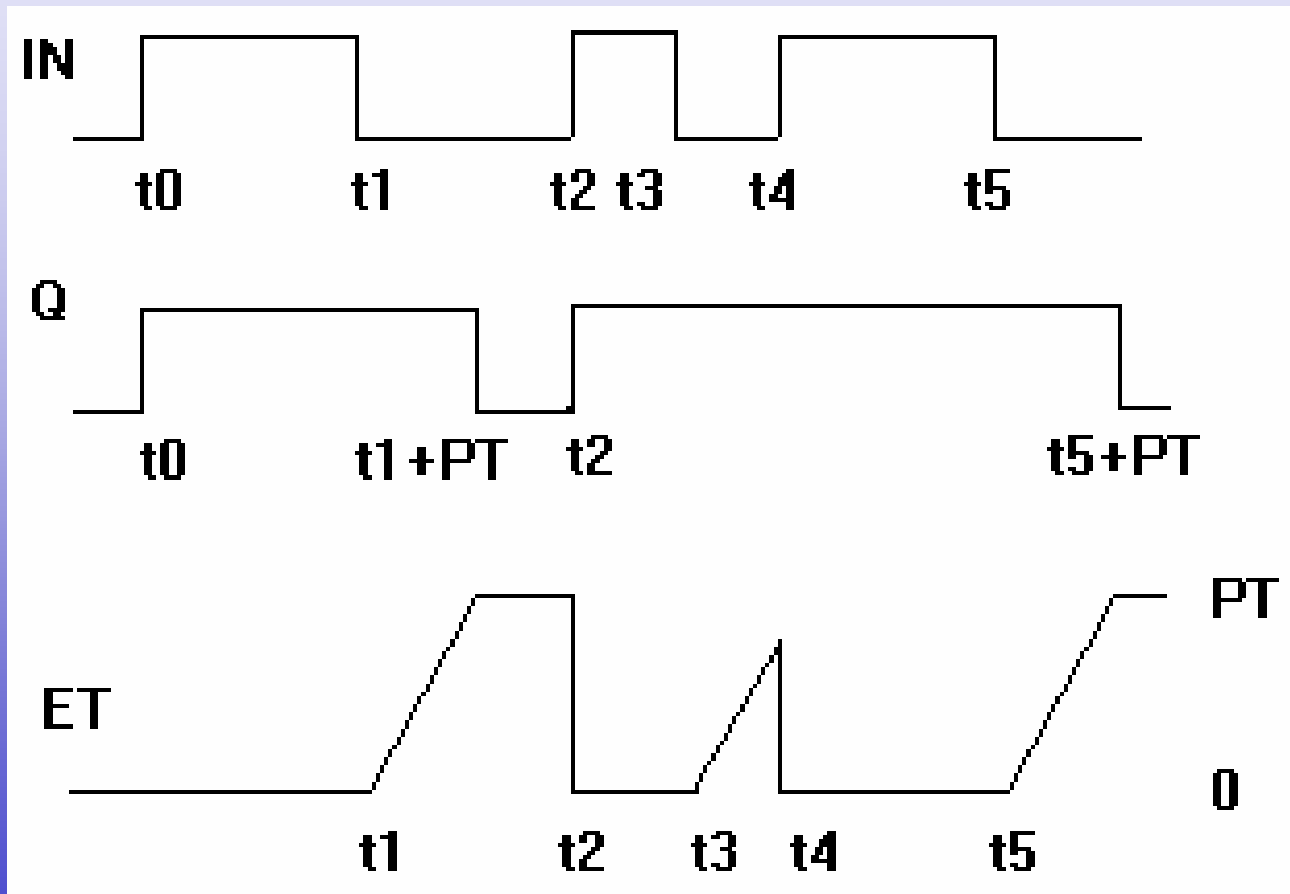
Таймер тип TP:



Таймер тип TON:



Таймер тип TOF:



Още някои използвани конструкции:

Масиви (arrays)

Пример: arr1 : ARRAY [1..5] OF INT := 1,2,3,4,5;

Структури (structures)

Пример: TYPE <Structurename>:
STRUCT
<Declaration of Variables>
END_STRUCT
END_TYPE

Номерирани типове (enumerations)

Пример: TYPE TRAFFIC_SIGNAL: (Red, Yellow, Green:=10);
END_TYPE

Присвоявания (alias) – създава специфичен за потребителя тип

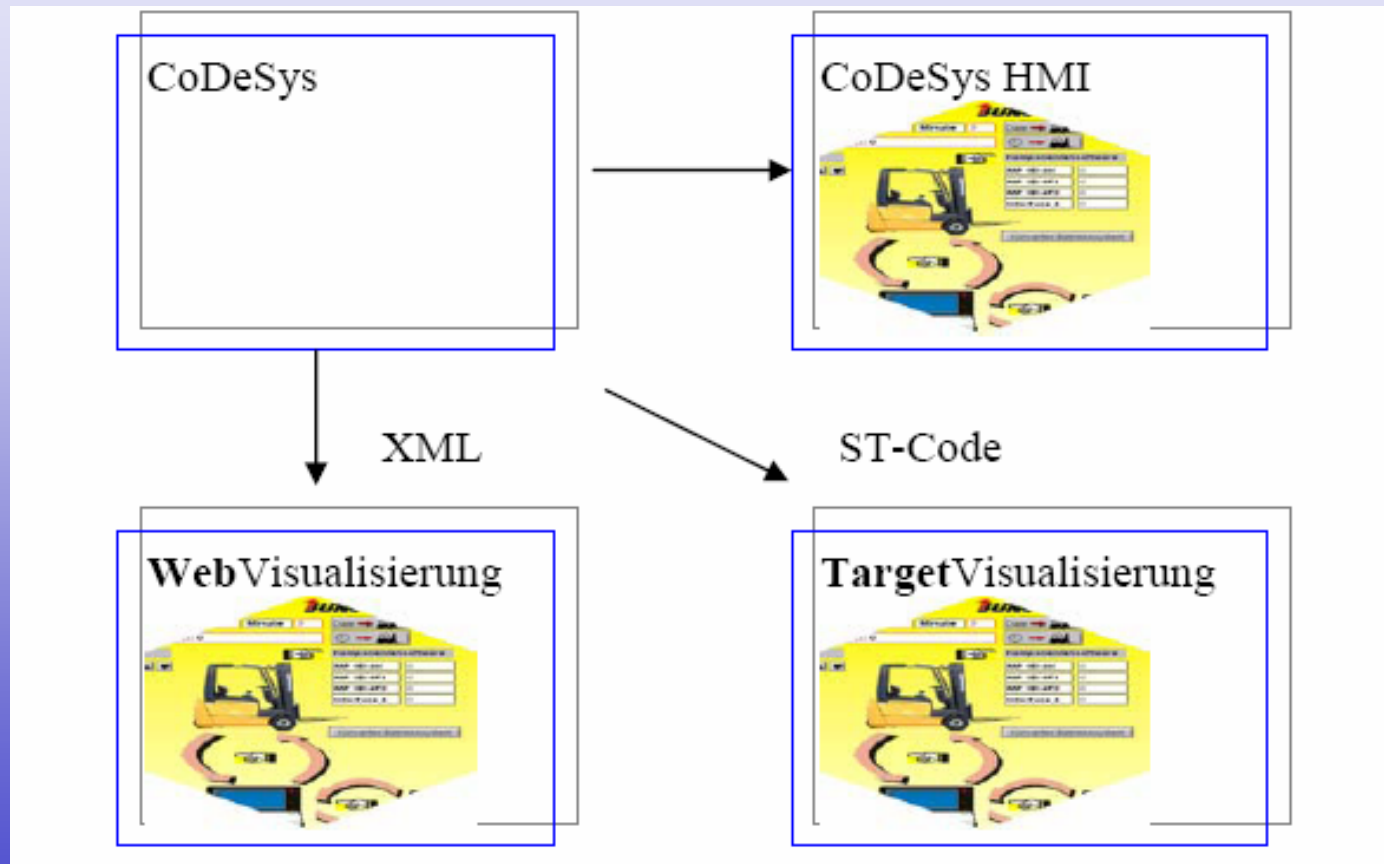
Пример: TYPE message:STRING[50];
END_TYPE;

PLControl

Симеон Трифонов



Потребителски визуализации:





AMK

Control your Motion.

PLControl

Край

Благодаря за вниманието!

маг. инж. Любомир Борисов

маг. инж. Симеон Трифонов

маг. инж. Мариян Няголов



Катедра “Автоматика,
информационна и управляваща
техника”