

Задание за софтуер на PHP, MySQL, HTML, CSS за управление на промишлени процеси чрез контролер OMRON

1. Хардуерна установка.

Стандартен компютър с инсталиран на него съвременен GNU/Linux операционна система. Инсталиран Апач, PHP и MySQL последните стабилни версии. Конфигуриран GNU/Linux в неговите startup и shutdown скриптове да се стартират и спират Апача, MySQL-а и SSH сървъри. Към компютъра по сериен порт е вързан промишлен микроконтролер OMRON.

2. Основни софтуерни изисквания.

Софтуера да се напише на PHP 5 без JavaScript, със валидиран HTML код като версия 4.01 Strict и висока. MySQL 5 база данни с таблици MyISAM или InnoDB по избор.

2.1 CSS

Визуалното форматиране се извършва от CSS файл който трябва да има във себе си две секции. Едната е за форматиране на HTML файла на екрана и една която е за форматиране на HTML файла при печат.

2.2 Строеж на HTML страницата

Една страница трябва да има <head> където в <title> тага динамично да се изобразява коя функционална част на софтуера се изпълнява. В <body> идва менюто с навигационните връзки и след това динамично променящото се съдържание. Най-накрая на страницата няма да има нищо след динамично извежданото съдържание.

Тука е времето да се помисли за сесийно (или друго) валидиране на потребители със име и парола. Сесията да се ограничи до 30 минути, след това да е необходимо отново да се логне с име и парола отделния потребител. Да има подразбиращ се потребител v-plus:v-plus

Да има файл за config.php в който да се описват основни настройки на софтуера и от там да се задава езика на интерфейса. Интерфейса се прави езиково независим.

3 Инсталиране на софтуера, правене на базата данни

Софтуера започва директно от кореновата директория на Апача. Ще има файл install.php който ще чете config.php и languages/bg.php (или en.php) за съответния език и ще направи базата данни. След това в index.php трябва да има проверка за съществуването на install.php и v-plus.sql и да дава грешка и да не визуализира други страници докато не се премахнат от кореновата директория на Апача.

4. Помощни, help файлове

install.txt е файл с обяснение как се инсталира софтуера и се намира в кореновата директория на Апача. license.txt съдържа лиценза за ползването на софтуера и се намира в кореновата директория на Апача. about.php е файл с обяснение как се работи със софтуера (скрийншоти, клавишни комбинации и др.), още съдържа информация за текущата версия на софтуера и връзка към license.txt

5. Компоненти

Компонентите трябва да могат да се създават, променат, трият.

Компонентите имат размерност, течка и уникално име

- трябва да има проверка за уникалност на име и течка

- трябва да има подсказка кои са свободни течки

Компонентите които вече са използвани веднъж в производството - пратени към контролера се "заклучват" за триене и редактиране.

- Времето няма течка и затова ще я сложим с някаква подразбираща се стойност и размерност, при самото създаване на базата данни ще е още първо начален флаг "заклучена компонента"

- при създаването на всички компоненти има флаг "отключена компонента" което дава възможност за триене и редактиране.

- Размерността предварително ще е зададена в HTML-а

- Проверка на входните полета

- Всеки ред вкаран в базата данни има timestamp и поле в което се вкарва username-а на потребителя извършил операцията

6. Рецепти

- създаване, променяне и триене на рецепти

- няколко компонента създават рецепта

- задължително трябва да присъства времето в рецептата

- Проверка на входните полета

- Всеки ред вкаран в базата данни има timestamp и поле в което се вкарва username-а на потребителя извършил операцията

7. Потребители

- създаване, променяне и триене на потребители
- Само v-plus:v-plus може да създава потребители
- v-plus може да променя паролите на всички и на себе си по всяко време
- всички потребители могат само на себе си да променят паролата
- Потребителите не могат да променат никаква друга своя информация
- v-plus може да променя и трие потребител и негови данни преди той да е "заклучен"
- "Заклучването" става когато потребител с неговото име започне работа със софтуера и направи запис

в базата данни, оттогава нататък само паролата може да му се сменя и допълнителната информация на потребителя.

- v-plus е заклучен още при създаването си
- ако 30 минути потребител не работи със софтуера трябва отново да се логне
- Всеки ред вкаран в базата данни има timestamp и поле в което се вкарва username-а на потребителя

извършил операцията

8 Видим интерфейс, с входни полета зависещи от оператор.

- Проверка на входните полета
- Всеки ред вкаран в базата данни има timestamp и поле в което се вкарва username-а на потребителя

извършил операцията

- Праграмата работи в 4 мерни единици:
= тегло -> килограм като инертни м-ли, вода и свързващи м-ли са в цели килограми а добавките са ->

0,00 кг

= време -> секунди

= обем -> кубични метри м3 (да се уточни колко знака след десетичната запетая 0,00?)

= разстояние -> километър

В софтуера има:

§ Номенклатури

§ Обекти

§ Камиони

§ Шофьори

§ Клиенти

§ Рецепти

§ Заявки

§ Влажности

Започва подробното представяне на подробности върху гореизброените раздели в софтуера.

§ Номенклатури

Номенклатурите от своя страна се делят на четири групи материали

- Инертни материали - 5 вида
- Свързващи вещества - 3 вида
- Вода - 2 вида
- Адаптиви (добавки) - 6 вида

- Инертни материали имат:

= id - int

= име - txt

= произход - txt

= зърнометрия - txt

= точка - int

- Свързващи вещества

= id - int

= име - txt

= произход - txt

= точка - int

- Водата бива два вида: вода и шлам

- Адаптиви (добавки)

= id - int

= име - txt
= произход - txt
= точка - int

§ Обекти имат:

= id - int
= име на обекта - txt
= адрес - txt
= разстояние - txt
= особености - txt

§ Камиони имат:

= id - int
= име на камион - txt
= номер на камион - txt
= обем в кубични метри (м3) - int

§ Шофьори имат:

= id - int
= име - txt

§ Клиенти

Клиентите от своя страна се делят на два вида (тука трябва да се подадат достатъчно данни за една фактура):

- фирми
= id - int
= име на фирмата - txt
= адрес - txt
= Ид. номер - int
= ДДС номер - int
= МОЛ - txt

- часни лица
= id - int
= име - txt
= ЕГН - int

§ Рецепти

= id - int
= време на замес (в секунди) - int (ограничено до 500)
= име на рецептата - txt
= клас по якост - txt
= клас по околна среда - txt
= клас по консистенция - txt
= сертификат - txt
= съставил: име - txt (128 символа)

* Рециптите се пишат винаги за 1 кубически метър.

** Схема на ред на изпълнение на рецептата (да се добави)

§ Заявки съдържа:

= Клиент - избира се от падащо меню
= Обект - избира се от падащо меню
= Обем - 000,0 м3, цяло число - кое е истина да се уточни?
= валидност ? (статус) - да се уточни тази позиция
= бутон [Запиши Заявката] -> отива се към следващата схема, нова страница

Избор на заявка за изпълнение

= Да се постави схемата тука
= бутон [Изпълни] -> води към прозорец за избор на шофьор и камион

Прозорец за избор на шофьор и камион

= Камион - избира се от падащо меню

- = Шофьор - избира се от падащо меню
- = Кубици на камиона []м3 - текстово поле, sql-ва се обема на камиона и се попълва там като стойност по подразбиране която може да се редактира от оператор
- = бутон [Изпълни] -> води към прозорец за въвеждане на влажностите и преизчисляване на заявката

Прозорец за въвеждане на влажностите и преизчисляване на заявката

- = Извежда таблица за номенклатурите по рецептата както е дадена за 1 м3
- = Извежда таблица за номенклатурите по заявка за []м3
- = Влага на инертните материали в %
- = бутон [Изпълни] -> отива към информационна страница където е призначена заявката според влажностите и показва колко бъркалки е заявката

§ Влажности

- = Влагата е процент от сухото вещество
- = 1200 кг / 1,01% = реални килограми сухо вещество
- = 1200 кг - реални килограми сухо вещество (сметнати по предната формула) = водата в сухото вещество
- = Следва тази "водата в сухото вещество" да се намали от общата вода
- = Трябва да се запази сухото вещество в такова количество каквото е по рецепта
- = След това се изчисляват колко бъркалки е заявката и се праща към контролера.

9 Комуникация с контролера

- = Писане - стойностите от изчислената заявка се ескейпват и се оформят двойки, стойност на номенклатурата*-клетка в паметта на контролера.

recepta_nomer	UINT	D100	nomer na receptata
recepta_tip	UINT	D101	tip na receptata
recepta_ime	CHANNEL	D102	chetiri dumi - 6 bajta
recepta_time_zames	UINT_BCD	D106	vreme za zames
recepta_zames	UINT	D107	broj zamesi
recepta_im_komponent1	UINT	D108	pari komponent na inertni materiali 5 komponenta
recepta_ciment1	UINT	D113	3 poredni dumi
recepta_voda	UINT	D117	napravna voda
recepta_shlam	UINT	D118	shlam
recepta_aditiv1	UINT	D119	pyrva dobavka onshto sa 6 edna sled druga
recepta_im_techka1	UINT	D125	pokazva koq techka teche
recepta_cimet1_techka	UINT	D130	
recepta_aditiv_techka1	UINT	D134	6 poredni dumi

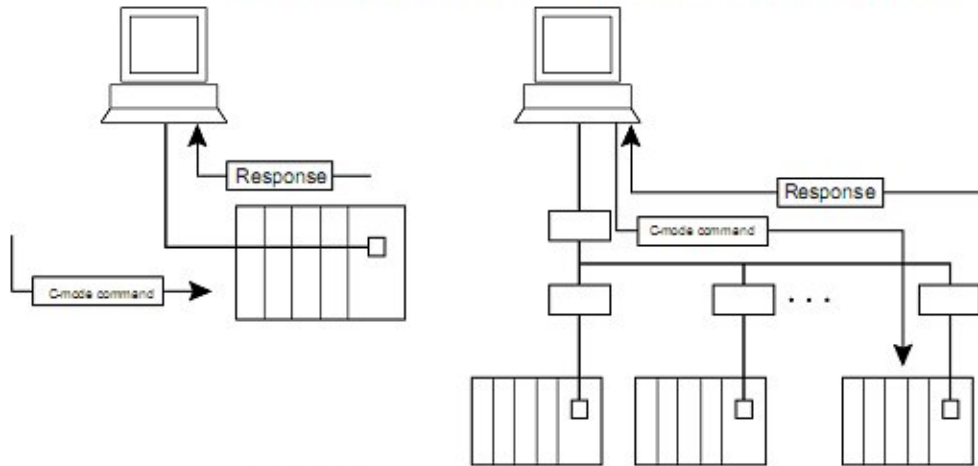
- = Четене - стойностите прочетени от контролера се записват от базата данни в row вид без изменение.

9.1 Формиране на комуникацията с контролера

- = Командите изпращани и приемани са до 1 фрейм големина – 131 байта
- = @00RD0000000157*.chr(13) (read data)
- = @00WD0000000157*.chr(13) (zapis data)
- = @00IC... check pdf (powerka na u-wo dali e zaka4eno)
- = 400-500ms timeout на комуникацията
- = да има три опита при което да дава грешка

2-1 C-mode Commands

C-mode (Host Link) commands form a command/response system for serial communications (Host Link Mode) to perform various control operations between a CPU Unit and a host computer directly connected to it. These operations include reading from and writing to I/O memory, changing operating modes, executing forced set and forced reset operations, and so on.



Unlike FINS commands, C-mode commands can only be addressed to a CPU Unit, and they cannot be used for message service outside of the local network. They cannot be used for functions such as file operations.

C-mode (Host Link) commands can be sent from a host computer connected to a CS/CJ-series Host Link Unit. Up to 32 PLCs (Host Link Units) can be connected to a single host computer. For identification, each Host Link Unit is assigned a unit number from 0 to 31.

The length of a single unit of a command or response exchange is called a "frame." A single frame contains a maximum of 131 characters of data. Characters are sent and received as ASCII.

Note For a CS/CJ-series PLC, a "Host Link Unit" can be the CPU Unit, a Serial Communications Unit, or a Serial Communications Board.

A maximum of 30 words of data can be transferred for the first command frame and a maximum of 31 words of data can be transferred for other command frames when reading or writing word data in I/O memory. When reading/writing more than 30 words of data, the data transfer will be processed in multiple transmissions, with 30 words in the first and up to 31 words in each of the following transmissions until the number of words set in the command has been processed.

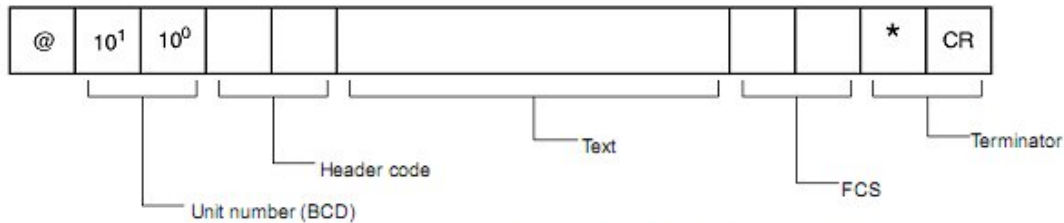
The frame formats for Host Link commands sent from a host computer and responses returned by the PLC receiving the commands are explained in the following section.

2-2 Command/Response Formats

Single-frame Commands

If a command is not more than 131 characters long, communications can be completed by sending a single command frame. This is called a "single-frame command."

Command Frame Format

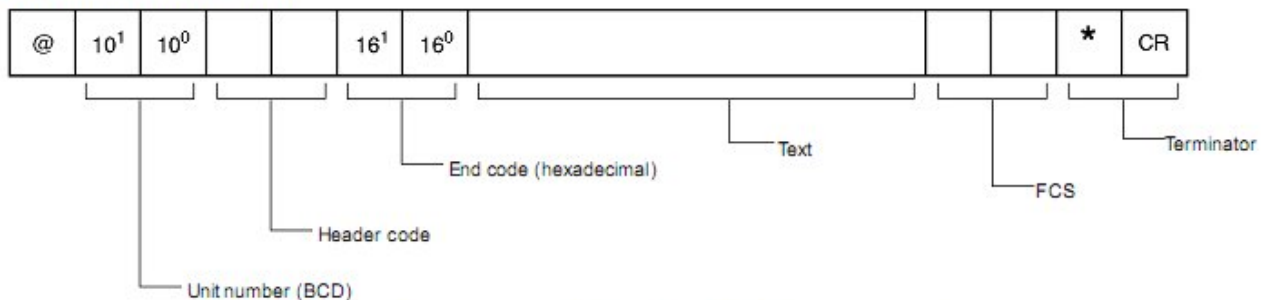


- @: Must be attached at the beginning of the command.
- Unit number: Set in BCD from 0 to 31 for each Host Link Unit.
- Header code: Specified in two characters.
- Text: Set parameters corresponding to command code.
- FCS: Calculate 2-character FCS (frame check sequence) at host computer. For details on calculating FCS, refer to *FCS Calculations* later in this section.
- Terminator: Set "*" and CR (CHR\$(13)) as two characters to indicate the end of the command.

Single-frame Response

If a response is not more than 131 characters long, the communications can be completed by returning one response frame. This is called a "single-frame response."

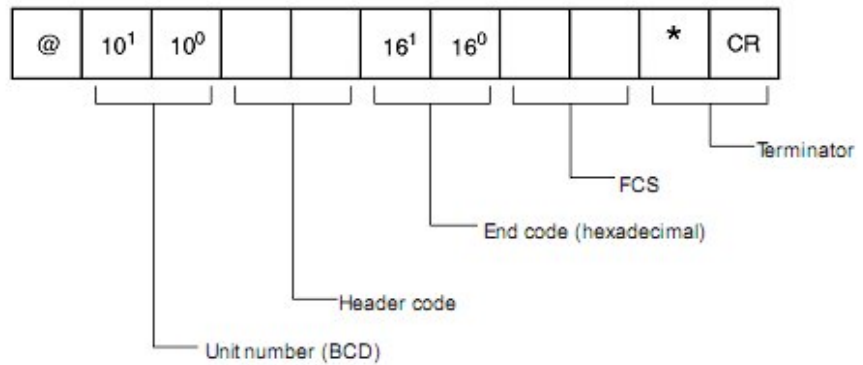
Response Frame Format



- @: Must be attached at the beginning of the response.
- Unit number: Set in BCD from 0 to 31 for each Host Link Unit.
- Header code: The command code that was received is returned.
- End code: The results (error status, etc.) of command execution is returned.
- Text: Returned only if there is read data.
- FCS: The 2-character FCS (frame check sequence) is returned.
- Terminator: Two characters indicating the end of the command, "*" and CR (CHR\$(13)), are returned.

Error Response Formant

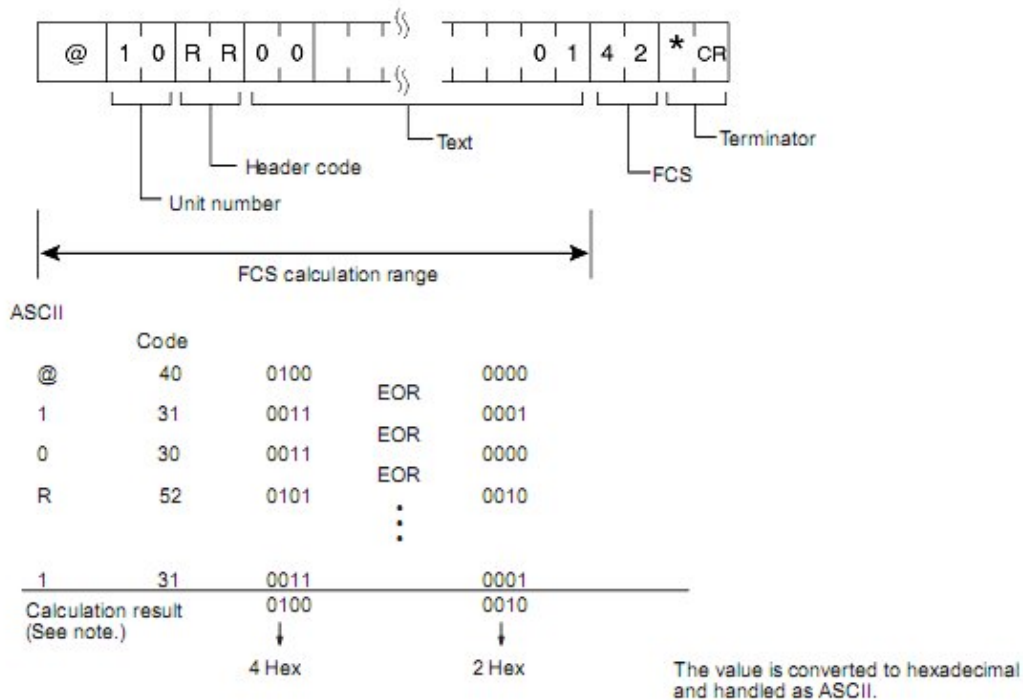
If a reception error or an error in executing the command occurs, a response is returned with no text.



FCS Calculations

The PLC calculates the FCS (Frame Check Sequence) value for each command frame it receives, and it checks for errors by comparing that value with the FCS value sent with the command frame. The host computer must calculate the FCS value when sending a command frame.

Also, when checking for errors in response frames, the host computer must calculate the FCS value for each response frame it receives and compare that value to the FCS value sent in the response frame.



Note The FCS is an 8-bit value converted into two ASCII characters. The 8-bit value is the result of an exclusive OR sequentially performed between each character in a transmission, from the first character in the frame to the last character of the text in that frame. Non-ASCII data, however, may sometimes be sent in the text data. If the data length is 7 bits, the leftmost bit of each character is masked before the FCS is calculated.