

Комплект драйвери за микропроцесорна система на база МК 68HC11

инж. Серафим Димитров Табаков
Т. У.-София

Abstract: This paper describes an input/output drivers system for Industry process control. The system is based on the MC68HC11 microcontroller. The main objectives of the drivers coding was to achieve easy portability and system independence as well as to be a part of a realtime multitasking system.

Разработката на системи за автоматизиране контрола и управлението на производствени процеси предполага наличието на средства за управление на входно-изходните им устройства. Обикновено тези системи се проектират на база едночипови 8 или 16 битови микроконтролери като например I8051 , MC68HC11(12/16) , Microchip PIC и други. Развойните системи за тях най-често включват крос-компилатори на С и асемблер. Стандартните библиотеки на С компилаторите включват известен брой функции за работа със входно-изходни канали и средства за управление на паметта. Неудобство представлява обаче факта , че голяма част от библиотечните функции с компилирани до обектни файлове готови за свързване в единна програма , които обаче са с прекалено големи размери. Така например при компилатора Whitesmiths 68HC11 С минималния размер на един библиотечен файл надхвърля 14 килобайта. Друго неудобство представлява липсата на драйвери за управление на стандартни входно-изходни устройства като например RS-232C , Centronics интерфейс , драйвери за управление на памети тип Flash и др.

Въпреки че една от основните функции на повечето промишлени системи базирани на микроконтролери е именно приемането и предаването на данни по различни видове най-често стандартни канали за връзка , в развойните системи липсват готови модули за обслужването им.

Предмет на доклада са разработените библиотеки драйвери за управление на входно-изходни устройства за едночипов микроконтролер МК68HC11 , като при разработването им една от основните цели беше да се направят възможно най-гъвкави , така че да не зависят от конкретната хардуерна реализация на системата. Втората цел при написването на драйверите, беше те да са максимално удобни за вграждане в многозадачни системи за реално време.

Един от начините за постигане на лесна преносимост на програмното осигуряване е максималното използване на системно независими езици от типа на C, C++ и др. Друг подход улесняващ работата на разработчика при създаването на софтуерни системи е свързан с обектното програмиране. Предпоставка за това е фактът че повечето от обектите на една система независимо от вида ѝ са почти или напълно еднакви. Така например управлението на сериен канал от типа RS-232 едва ли ще зависи от това дали този интерфейс е част от персонален компютър или от промишлен контролер. Наличието на готов драйвер за управлението на този обект подобен на използваните такива в операционните системи и обектноориентираните езици от високо ниво за тях при персоналните компютри напримерно би улеснило разработката и настройката на програмите.

Поради тази причина конструкцията (и използването) на голяма част от разработените драйвери е близка до драйверите за управление на същите обекти в системите за обектно програмиране като например Visual C и Visual Basic.

Пример за това може да бъде посоченият по-горе интерфейс RS-232. На фиг.1 е показана логическата структура и алгоритмите за управление на канала при приемане (фиг.1-а) и при предаване (фиг.1-б) в псевдо-асемблерен формат. След стартирането на системата се инициализират част от параметрите на канала ,например честота на предаване и приемане , брой битове в предавания байт ,брой стоп-битове , големина на приемния и предавателния буфер и др. При наличие на на входна информация (генерирано прекъсване) тя се записва в приемния кръгов буфер на интерфейса , като системата би могла да прочете в даден момент от време както цялата пристигнала информация , така и само част от неяили количеството пристигнали байтове например. При предаване ,информацията се записва в предавателния буфер и първият байт от него се прехвърля в предавателния регистър на интерфейса. След това предаването на следващите байтове става по прекъсване т.е. е “в сянка” за разработчика ,като е възможно генерирането на събития от типа “празен предавателен буфер” или “предаден байт” при многозадачните системи за реално време.

В посочения пример приемането и предаването на информация е направено на база използване механизма на прекъсвания генерирани от хардуера на конкретния интерфейс. Това обаче не винаги е възможно поради факта че не всички входно-изходни устройства имат подобен механизъм. Типичен пример за това е извеждането на символна или графична информация. Обикновено за целта се използват различни видове светодиодни индикатори, течнокристални панели, плазмени или вакуумнолуминисцентни индикатори. При многозадачните системи системи би могъл да се реализира механизъм за подаване на заявка за отразяване на

текущата информация чрез използване на пощенската кутия на задачата изпълняваща тази функция. Ако обаче операционната система на контролера не е проектирана като многозадачна то при изобразяването на информация най-добре би било да се използват дадените примитиви на езика (напр. printf(), putchar () или putchar() при използването на C), като се пренапишат за конкретния вид дисплей. Напоследък едни от най-често използваните дисплеи за изобразяване на буюено/цифрова информация са течнокристалните - това се дължи както на намаляване на цената им , така и на удобствата които предлагат - ниска консумация , използване на ASCII кодове и най-вече поддържането на стандартни команди за управлението им независимо от фирмата производител за повечето от тях. Последното именно дава възможност за написването на преносими управляващ драйвер за многозадачна система или модифицирани примитиви за езика C.

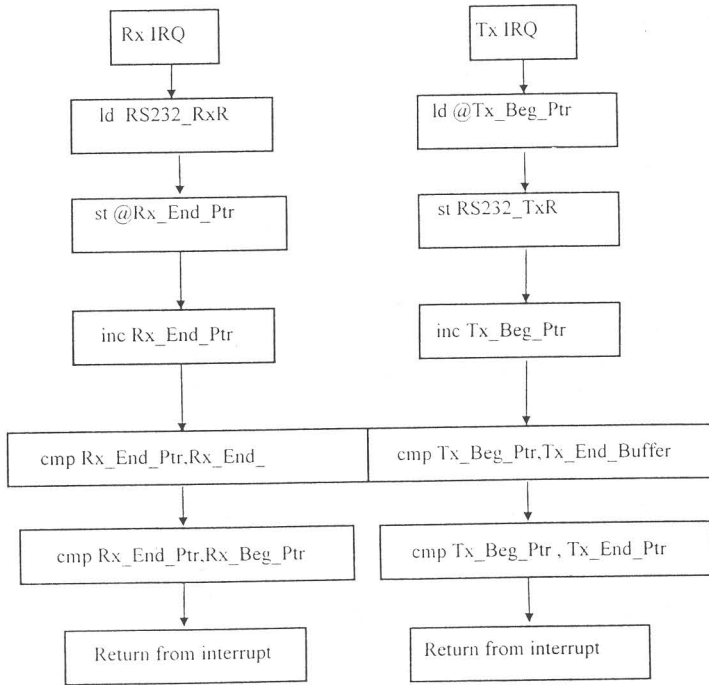
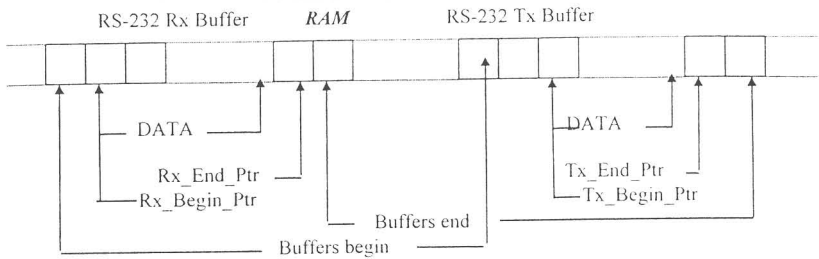
Освен показаните дотук , предмет на разработка бяха още драйвер за емулиране на I²C master интерфейс , на вградения в едночиповия микроконтролер SCI , драйвер за управление на паралелен интерфейс тип Centronics , и система от подпрограми за работа с Flash памети както и възможност за запис на данните от паметта на 3,5” дискета с поддържане формат на файловете и файлова система съвместими с MS-DOS(фиг.2). При написването им е използван пакета крос компилатор и асемблер Whitesmiths

Предстои разработка на драйвери за връзка с клавиатура от PC и за управление на вградения в микроконтролера SPI .

Всички представени в доклада драйвери са вградени във внедрен промишлен микроконтролер за събиране на данни.

Литература :

“MC68HC11 - Technical Data” - MOTOROLA , Literature Distribution Center



a)

b)

fig.1

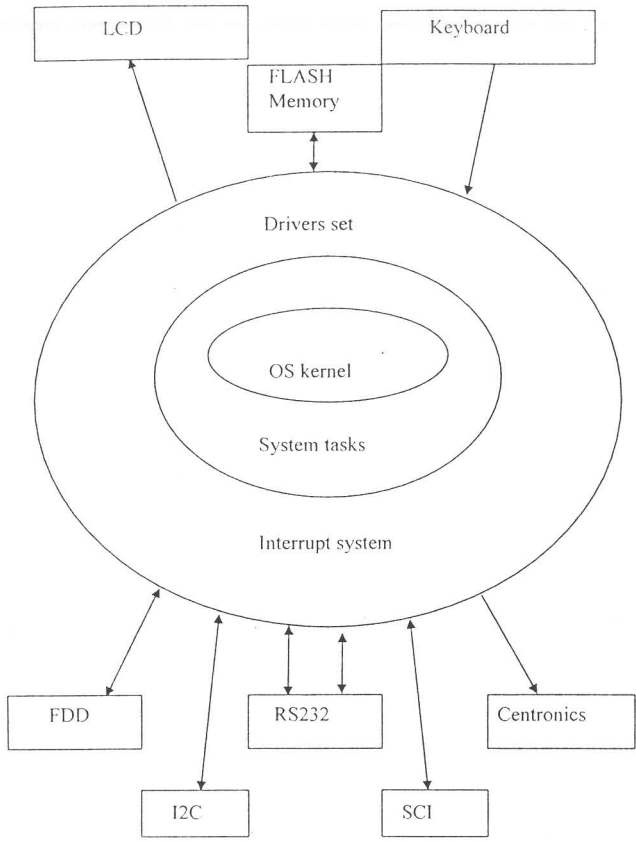


fig.2