При роботі з мікроконтролерами сімейства PIC, який був обраний для керування апаратною частиною системи, для написання програм керування існує можливість використання лише двох мов програмування: мови низького рівня асемблера та мови високого рівня С.

Програма розроблена на мові асемблера виконується швидше та потребує менше ресурсів ніж аналогічна програма написана на С. Написання програми для мікроконтролера на асемблері дозволяє краще взаємодіяти із вбудованими апаратними засобами мікроконтролера.

Іще однією превагою використання асемблера на етапі розробки є те, що деякі середовища схемотехнічного моделювання, наприклад Porteus ISIS, дозволяє використовувати не тільки скомпільовані програми, а і вихідний текст програми на асемблері. Це може суттєво прискорити розробку простих рішень та відмовитись від встановлення спеціальних інтегрованих середовищ розробки.

Однак за таку оптимізацію доводиться платити більшими затратами часу на розробку програми. Оскільки для розробки на асемблері необхідно добре розуміти архітектуру конкретного мікроконтролера, для якого ведеться розробка.

Слід також зауважити, що асемблер для мікроконтролерів PIC значно відрізняється від звичного асемблера архітектури x86. Більше того кожна серія мікроконтролерів має власну реалізацію мови асемблера. Проте фірма Microchip оснащує свої продукти детальною документацією в якій описано як апаратні особливості архітектури так і про особливості взаємодії з ними, тому досвідченим розробникам апаратури використання асемблера не завдасть значних труднощів.

Іншим можливим рішенням є використання мови С, переваги якої складно переоцінити. На відміну від асемблера мова С більш зрозуміла для розробники та дозволяє в деякій мірі абстрагуватись від апаратних особливостей архітектури мікроконтролера. Що в свою чергу зробить вихідний код більш гнучким та спростить його перенесення на інший тип мікроконтролера. За роки існування цієї мови в інтернеті накопчелась велика кількість алгоритмів для рішення майже будь-якої задачі. Тому більшість розробників віддають перевагу саме цій мові.

Однак програмування на С потребує використання спеціальних інтегрованих середовищ розробки. На щастя фірма Microchip поставляє таке середовище, що розроблене спеціально для мікроконтролерів PIC.

Таким середовищем є MPLAB фірми Microchip. Він має набір інструментів для розробки та налагодження програмного коду для керування мікроконтролерами, такі як компілятори, лінкувальники, засоби для слідкування за ходом виконання коду. Для кожної серії мікроконтролерів необхідно використовувати різні компілятори, тому MPLAB створений з окремих модулів та дозволяє встановлювати різні інструменти, залежно від ситуації.

За основу взято демонстраційний приклад

Тактова частота для роботи ЮСБ 5МГц

Таймери та переривання

Керування напругою колектор-емітер відбувається за допомогою широтно-імпульсної модуляції (ШІМ), тобто результуюча напруга залежить від шпаруватості імпульсів що подаються на вхід схеми керування. Це створює необхідність у відносно високій стабільності генерованих імпульсів, яку не можна забезпечити при керуванні з основного циклу програми. В цьому випадку можна скористатись вбудованими таймерами мікроконтролера.

Використання таймерів, як і більшості периферійних модулів засновано на перериваннях. Переривання це механізм, що дозволяє перервати виконання програми та передати керування іншій ділянці коду. По завершенню виконання цієї ділянки керування буде передано туди де було перервано виконання основного коду. Це дозволяє швидко реагувати на настання певних подій.

Мікроконтролери сімейства PIC18 мають дворівневу систему переривань, тобто переривання можуть мати два пріоритети: високий і низький, відповідно і дві функції обробники. [<http://pro-diod.ru/programms/pic-micro/interrupt-v-pic18.html>]

Компілятор мови С надає досить зручний засіб для роботи з обробниками переривань. За документацією, щоб визначити функцію як обробник необхідно додати ключове слово *interrupt*, та за необхідності пріоритет (наприклад *low\_priority*) у визначення функції, яке матиме наступний вигляд: *void interrupt low\_priority  low\_isr (void)*. Це означає що функція буде викликана при появі переривання з низьким пріоритетом.

На жаль в нових версіях компілятора таке визначення обробників переривань не працює. Однак адреси потрібних функцій можна «вручну» занести у відповідні вектори переривань. Вектор — це адреса в програмній пам'яті, за якою відбувається автоматичний перехід при виникненні будь-якого дозволеного переривання.

Як було зазначено даний мікроконтролер має два типи переривань, а отже і два вектора:

* 0x08 — вектор переривань з високим пріоритетом;
* 0x18 — вектор переривань з низьким пріоритетом.

Слід пам’ятати, що вразі відключення пріоритетів то буде використовуватись лише один вектор 0x08.

Щоб налаштувати переривання необхідно дозволити переривання глобально та периферійні встановленням біті PEIE та GIE регістру INTCON відповідно та пріоритети встановленням IPEN регістру RCON.[<http://subscribe.ru/archive/comp.soft.prog.pic/200703/13210529.html>]

ШІМ

Найкращий варіант роботи керованого перетворювача постійного струму на основі широтно-імпульсної модуляції це використання апаратних засобі ШІМ (PWM, Pulse-Width Modulation) мікроконтролера. Для цього в мікроконтролері передбачено модуль CCP. Абревіатура розшифровується як Capture/Compare/PWM (Захват/Порівняння/ШІМ). Обрана модель мікроконтролера має два незалежних модулі CCP1 та CCP2. Однак вони, є не зовсім незалежними, оскіьки працюють на одній частоті та використовують однаковий спосіб підключення, проте є можливість задавати шпаруватість для кожного з них незалежно.

Для роботи модуля потрібно налаштувати виводи RC2 (CCP1) та RC1 (CCP2) на вихід, та активувати таймер 2. При використанні модулів CCP цей таймер буде рахувати такти та керувати довжиною імпульсів, а отже не доступний для інших функцій.[<http://picdevices.ru/eksperimentyi/pwm-dac-lpf-ili-kak-eto-po-nashemu-shim-tsap-fnch.html?doing_wp_cron=1464891333.4888598918914794921875>]

Ініціалізація модуля CCP режимі ШІМ включає наступні етапи. Перш за все потрібно встановити режим роботи модуля та встановити відповідні виводи порту С на вихід. За режим роботи відповідають біти 0–3 регістру CCP1CON. Вибір режиму ШІМ відбувається встановленням бітів 2 та 3 (останні два ігноруються і можуть мати будь-яке значення).

За період імпульсів відповідає регістр PR2, що має 8 розрядів, а отже його значення не може перевищувати 255. Фактичне значення періоду розраховується за наступною формулою:

Ppwm = (PR2 + 1) \* 4 \* Tosc \* Nprescale,

де Ppwm — період імпульсів;

Tosc — тактова частота роботи мікроконтролера;

Nprescale значення переддільника таймера. Цей параметр може приймати три значення 1, 4 або 16. Для цього в біти 0–1 регістра T2CON потрібно встановити в значення 00, 01 або 11 відповідно. Використання регістра PR2 та таймера можна побачити на рисунку

Рисунок

За формулою на основі потрібної частоти імпульсів можна розрахувати значення PR2. Краще спершу взяти значення переддільника рівним 16 та розрахувати PR2, якщо значення перевищить 255 то взяти менше значення переддільника та повторити розрахунок.

Отже значення регістру PR2 та переддільника таймера впливають на період імпульсів