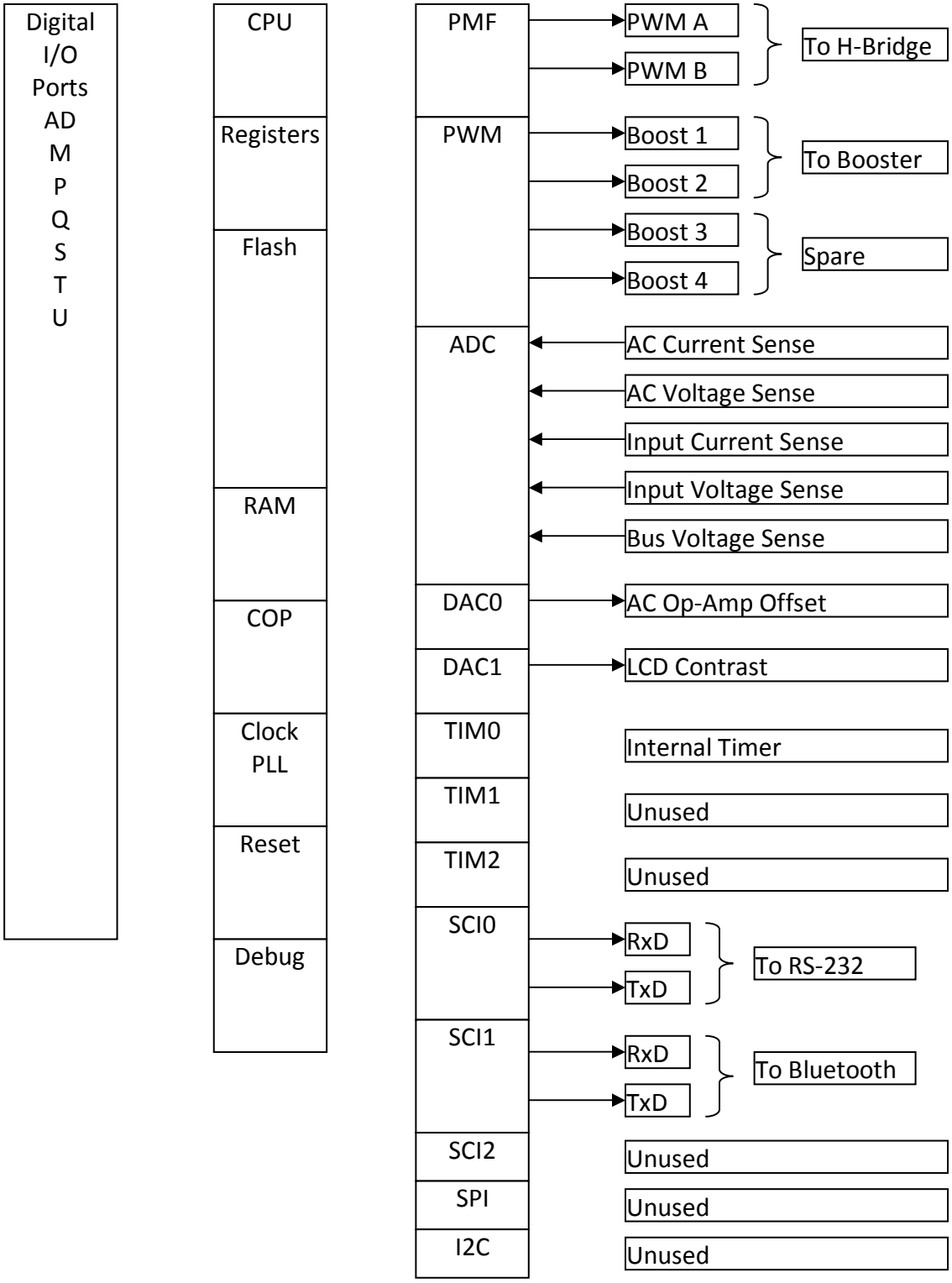
Microcontroller and Compiler

The Freescale MC9S12E128 microcontroller is a 16-bit processor integrated with flash EEPROM, RAM, and a large number of peripheral circuits. A few years ago these would all be individual IC modules. Now they are contained in a single chip but have to share a limited number of external pins.



Setting-Up to Write Software

The 1st step in writing software for this micro is to download and install the free compiler from Freescale.

Go to: Freescale Web Site.
https://www.freescale.com/webapp/sps/site/overview.jsp?code=CW_SPECIALEDITIONS

Download and install: Integrated Development Environment (IDE) and Compiler.
Special Edition: CodeWarrior for HCS12(X) Microcontrollers (Classic)

Load: After starting the IDE, load the project file.
CodeHC9S12.mcp

A tool for transferring object code into the development system is required. One possibility is:
USB Multilink Universal from,
http://www.pemicro.com/products/product_viewDetails.cfm?product_id=15320137

Address Spaces

Registers

Registers occupy the low addresses of memory space.
Address: 0x0000 – 0x03FF

RAM

After reset the RAM overlaps the register space. That is:
Register Space: 0x0000 – 0x03FF
RAM Space: 0x0000 – 0x1FFF
Since register space has priority over RAM, only 0x400 – 0x1FFF of RAM is available.

Modifying the Start-Up File

In order to recover all of precious RAM, it must be moved to a different location. It can be fixed to occupy 0x2000 – 0x3FFF where all of it is available to software.

The simplest way to implement this move is to add the following statement to the start-up file supplied by CodeWarrior when a project is first set-up.
In Start12.c add the following:
#define _HC12_SERIALMON

This will invoke the following supplied code:
#if defined (_HC12_SERIALMON)

```
        /* for Monitor based software remap the RAM & EEPROM to adhere
           to EB386. Edit RAM and EEPROM sections in PRM file to match these. */
        ___INITRG = 0x00;  /* lock registers block to 0x0000 */
        ___INITRM = 0x39;  /* lock Ram to end at 0x3FFF */
        ___INITEE = 0x09;  /* lock EEPROM block to end at 0x0fff */
#endif
```

The Project.prm file also has to be modified as follows:
RAM = READ_WRITE 0x2000 TO 0x3FFE;
Notice that the last address is 0x3FFE not 0x3FFF. This appears to be a bug in CodeWarrior that it complains of an overlap due to the stack being set to the last RAM address as:
STACKTOP 0x3FFFF.

Flash

This project will use only the direct flash pages.
Address: 0x4000 – 0x7FFF
Address: 0xC000 – 0xFFFF
Vectors: 0xFF80 – 0xFFFF

Peripherals

This microcontroller contains a rich set of peripherals. The following table lists the peripherals that are used by this application.

PMF – Pulse Width Modulator with Fault Protection

PW00	-	PWM_A_DRIVE
PW01	-	PWM_B_DRIVE
PW02	-	I/O
PW03	-	I/O
PW04	-	Unused
FAULT0	-	I/O
FAULT1	-	I/O
FAULT2	-	I/O
FAULT3	-	I/O
IS0	-	I/O
IS1	-	I/O
IS2	-	I/O

PWM – Pulse Width Modulator

PW10	-	BOOST_PWM_A
PW11	-	I/O
PW12	-	BOOST_PWM_B (Spare)
PW13	-	I/O
PW14	-	Unavailable on 80-pin package
PW15	-	Unavailable on 80-pin package

ADC – Analog-to-Digital Converter

AN0	-	AC_CURRENT_SENSE_MICRO
AN1	-	AC_VOLTAGE_SENSE_MICRO
AN2	-	INPUT_CURRENT_SENSE_MICRO
AN3	-	INPUT_VOLTAGE_SENSE_MICRO
AN4	-	BUS_VOLTAGE_SENSE_MICRO
AN5	-	Spare
AN6	-	Spare
AN7	-	Spare
AN8	-	I/O
AN9	-	I/O
AN10	-	I/O
AN11	-	I/O
AN12	-	I/O
AN13	-	I/O
AN14	-	I/O
AN15	-	I/O

DAC0 – Digital-to-Analog Converter 0

DAO0	-	Voltage offset for AC voltage and current op-amps

DAC1 – Digital-to-Analog Converter 1

DAO1	-	LCD contrast control voltage

TIM0 – 4 Timer Channels

Channel 4 used as internal 1 ms timer. All external pins assigned to I/O.

TIM1 – 4 Timer Channels

Unused circuit: all external pins assigned to I/O.

TIM2 – 4 Timer Channels

Unused circuit: all external pins assigned to I/O.

SCI0 – Serial Communications Interface 0

RxD0	-	Receive data from RS-232 receiver.
TxD0	-	Transmit data to RS-232 driver.

SCI1 – Serial Communications Interface 1

RxD1	-	Receive data from Bluetooth transceiver
TxD1	-	Transmit data to Bluetooth transceiver

SCI2 – Serial Communications Interface 2

Unused: all external pins assigned to I/O

SPI – Serial Peripheral Interface

Unused: all external pins assigned to I/O

IIC – Inter-Integrated Circuit

Unused: all external pins assigned to I/O