



Freescale Semiconductor, Inc.

*M68HC08
Microcontrollers*

*Direct Current Ignition
Reference Design*

*Designer Reference
Manual*

DRM011/D
Rev. 0.0, 3/2003

MOTOROLA.COM/SEMICONDUCTORS



Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

**For More Information On This Product,
Go to: www.freescale.com**

Direct Current Ignition Reference Design

Designer Reference Manual — Rev 0

by: Kenny Lam
Motorola Ltd
Hong Kong



Table of Contents

Designer Reference Manual — Table of Contents

Section 1. Direct Current Ignition

1.1	Introduction	7
1.2	Overview	7
1.3	Hardware Descriptions	8
1.4	Firmware Description	10
1.5	Extra Features	19
1.6	Further Information	21

Section 2. Glossary



Section 1. Direct Current Ignition

1.1 Introduction

This manual describes a reference design of a Digital Direct Current Ignition for Motorbike by using the MC68HC908JK3.

For the full MC68HC908JK3 specification, please refer to the data sheet, Motorola order number: MC68HC908JK3/D.

1.2 Overview

The Motorola MC68HC908JK3 is a member of the HC08 Family of microcontrollers (MCUs). It is a 8-bit low cost general purpose MCU with 16-bit free run timer. The features of the MC68HC908JK3 include input capture and output compare functions for real time control. The MC68HC908JK3 is available in several packages to fit into various applications. The main features of MC68HC908JK3 in this application include:

- Low cost design
- Precise HT coil charging / discharging time controlled by Output Compare function
- Automatic switch to and from analogue to digital ignition timing control
- Able to expand for Multi-ignition curves against analogue throttle (ADC)

Direct Current Ignition

1.3 Hardware Descriptions

Freescale Semiconductor, Inc.

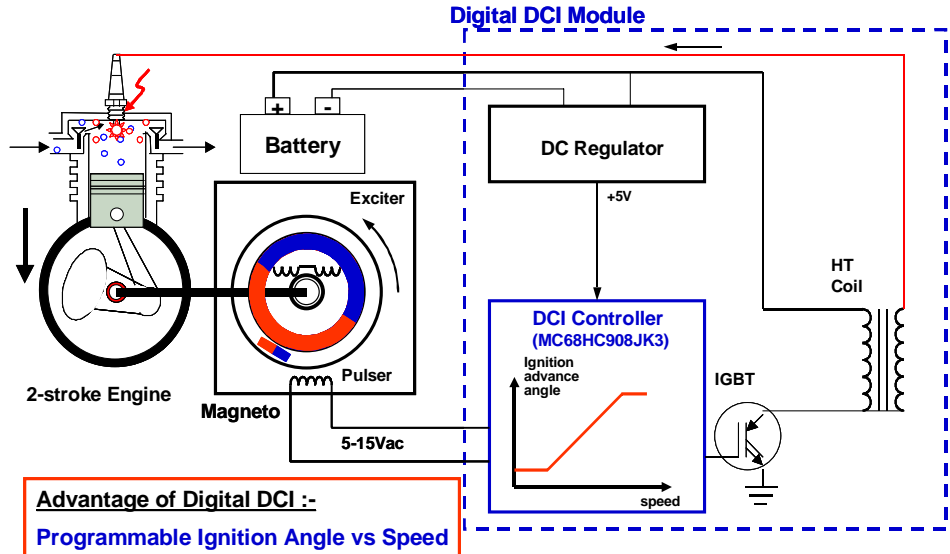
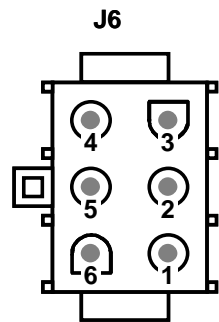


Figure 1-1. Block Diagram



Pin	Signal Name	Type	Voltage
1	PULSER_P	input	-15V to 15V
2	IGNITION	output	0 to 400V
3			
4	+SUPPLY	Supply	+12V
5	GND		
6	PULSER_N	input	-15V to 15V
<i>(Note:- PULSER_P & PULSER_N are isolated signals from GND.)</i>			

Figure 1-2. Module Connections

Figure 1-1 shows the block diagram of the digital DCI. The solution includes the pulser input from magneto, HT primary coil connection and

voltage supply regulator. The connections of the corresponding signal pins for DCI module are shown in [Figure 1-2](#). The battery supply 12V, ground, pulsers and HT coil connection pins respectively.

[Figure 1-3](#) shows the DCI printed circuit board.

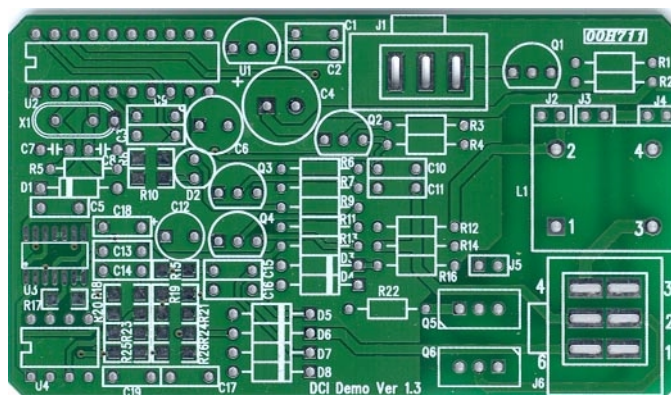


Figure 1-3. DCI PCB

- J1 can be used for Throttle input
- J2 and J4 are to control bypassing the L1 coil
- J3 controls the Tacho on/off output
- J5 connects solenoid valve (B/S) to control the air intake to cylinder
- J6 is the socket of the DCI module, shown on [Figure 1-2](#)
- D2 (LED) is the indicator for analogue ignition on/off
- Q1 works as the buffer to drive Tachometer
- Q2 is working for digital throttle input
- Q3 is high side driver for ignition
- Q4 is the low side driver to control analogue ignition on/off
- Q5 is the high current low side driver to control solenoid on/off
- Q6 is the high power (high voltage high current, IGBT) for charging/discharging the HT coil for ignition

Direct Current Ignition

1.4 Firmware Description

The firmware is divided into 2 portions, which are foreground and background jobs.

- Foreground job is to serve the system initialization, such as, RAM initialization, pointer setup, I/O PORT setup and some calculations (not time critical).
- Background job is to serve time critical sub-routines, such as, pick up coil input signal processing, output compare to ignition precisely, timer overflow, etc. They are high priority jobs. All of them are time critical tasks.

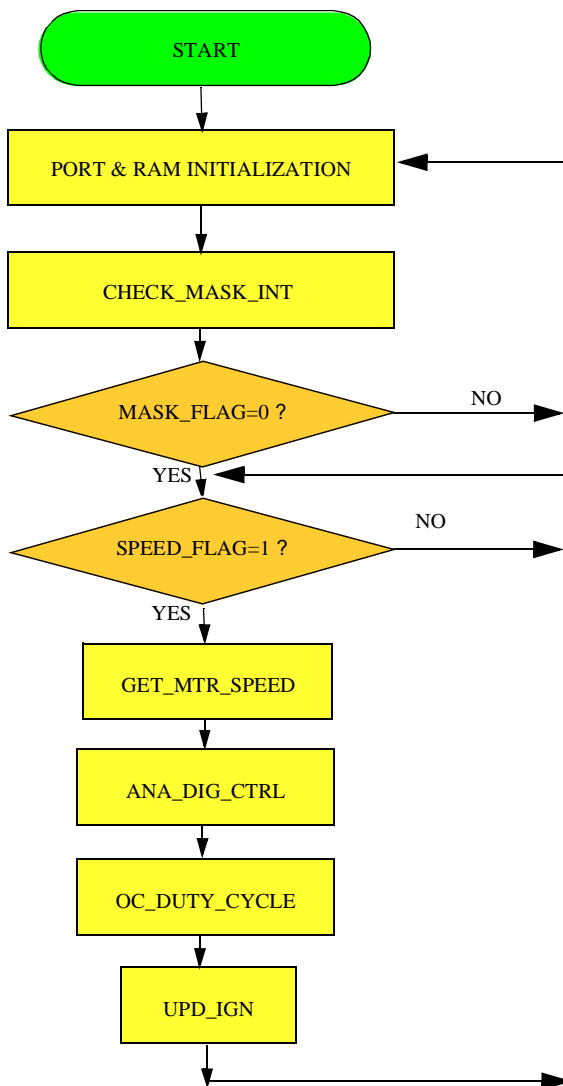


Figure 1-4. Foreground

1.4.1 CHECK_MASK_INT

This subroutine is to check whether the engine is running on idle. The MASK_FLAG is set when the engine is idle, otherwise, the MASK_FLAG is clear.

The SPEED_FLAG is set inside the IRQ_INT interrupt subroutine if the engine speed can be recognized by the MCU, otherwise, it goes back to the CHECK_MASK_INT subroutine.

Direct Current Ignition

1.4.2 GET_MTR_SPEED

This subroutine is to calculate the engine speed, which is the result of the subtraction of INT_PRE_CNT from INT_CUR_CNT.

1.4.3 ANA_DIG_CTRL

This subroutine is to select the ignition type (digital or analogue) depending on the engine speed. The D2 (LED) is an ignition type indication. It turns on the D2 (LED) if the engine speed is running less than 960rpm to indicate the ignition timing is being controlled by the analogue circuit. It turns off the D2 (LED) if the engine speed is running greater than 960rpm to indicate the ignition timing, which follows the curve defined by engine manufacturer, is being controlled by the MCU.

1.4.4 OC_DUTY_CYCLE

This subroutine is to calculate the charging time and discharging time for the HT coil and setup the necessary variables for the output compare function.

1.4.5 UPD_IGN

This subroutine is to calculate the ignition timing and setup the necessary variables in the output compare function for digital ignition.

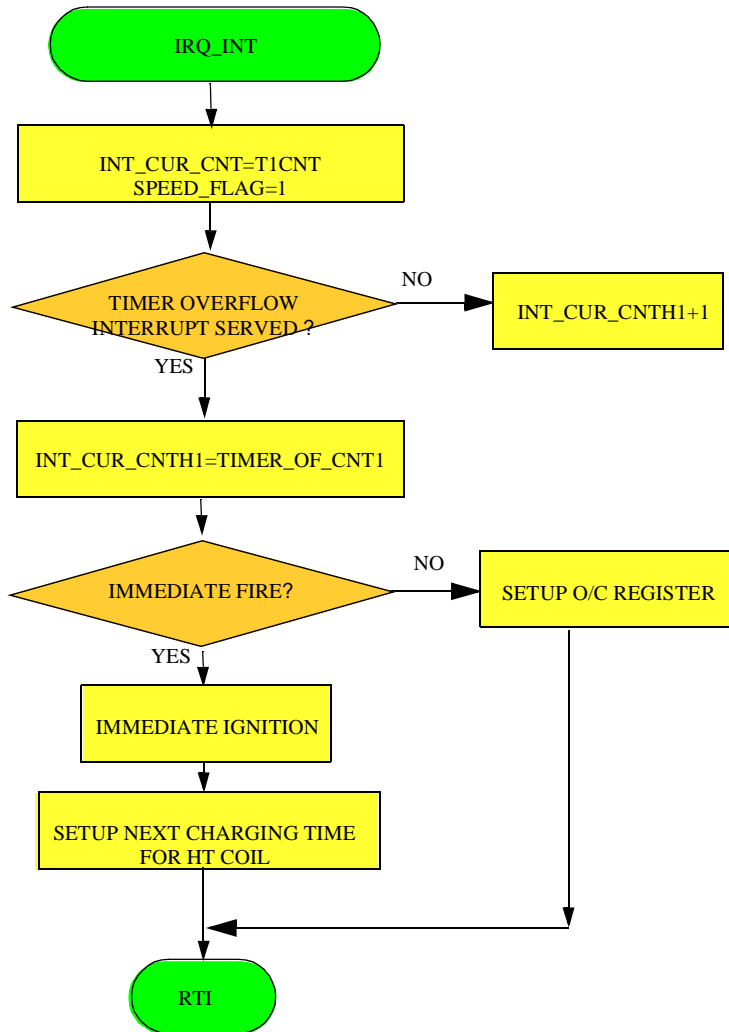


Figure 1-5. Background (IRQ_INT)

1.4.6 IRQ_INT Routine

Figure 1-5 shows the IRQ service subroutine flowchart. The subroutine updates the internal variables and sets up the output compare (O/C) registers where necessary for ignition timing interrupts.

Direct Current Ignition

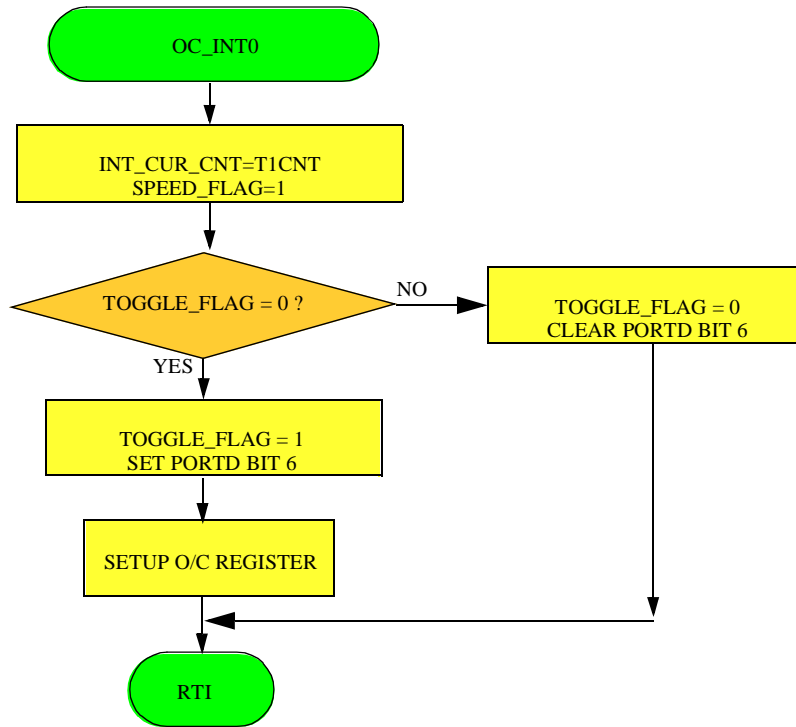


Figure 1-6. Background (OC_INT0)

Figure 1-6 shows the output compare interrupt service subroutine. It automatically responds to the output compare register (T1CNTH/L) in order to control the HT coil charging/discharging time.

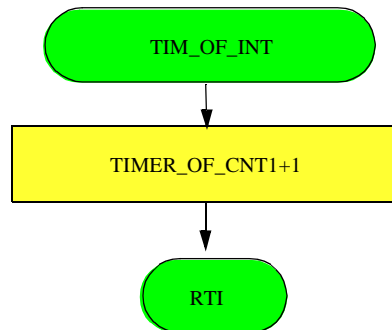


Figure 1-7. Background (TIM_OF_INT)

Figure 1-7 shows 16-bit timer overflow interrupt service routine. It handles the conversion from 16-bit to 24-bit counter.

1.4.7 Software Testing

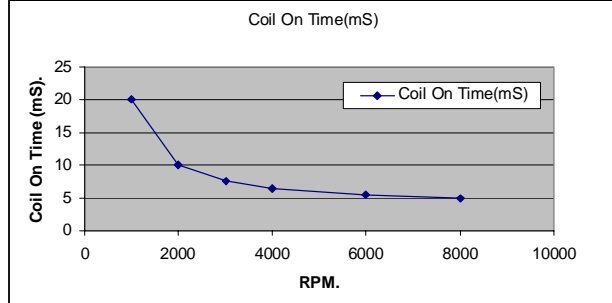
The following equipment or equivalent is required for the software validation.

Use an HP 33120A Signal Generator to generate a 5V single pulse with frequency range from 10Hz (600 rpm) to 150Hz(9000 rpm) to simulate the real engine speed for software test. It directly connects to the JK3 IRQ pin to simulate the engine speed detection.

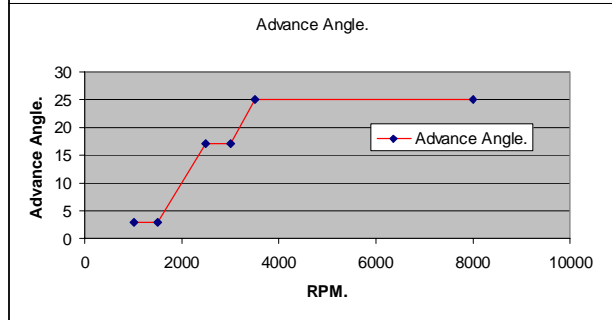
- Use a Tektronix TDS 540 digital scope to capture the signal input from the JK3 IRQ pin and the signal output from the JK3 port D bit 6 pin.
- The relationship between the input IRQ (RPM) and the output port D bit 6 (HT coil charging and discharging time for ignition) is shown on the next page.

HT coil Charging time and ignition time requirements

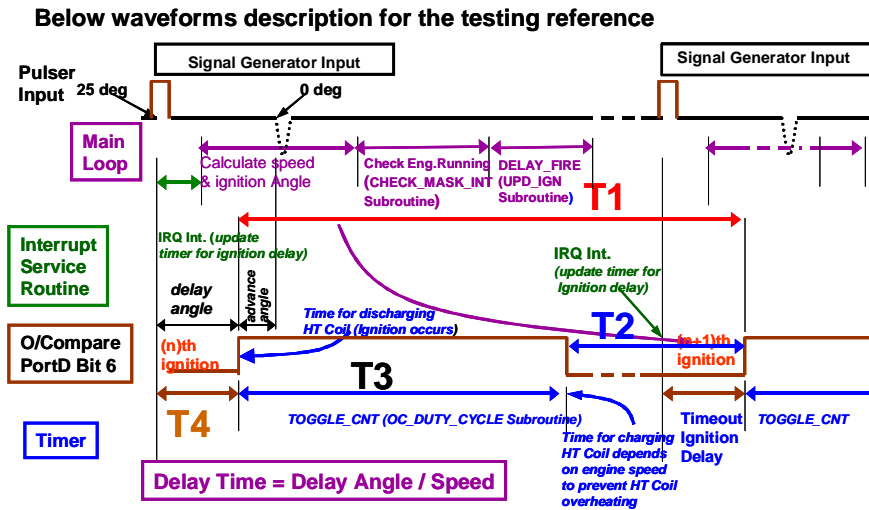
RPM	Coil On Time(mS)
1000	20
2000	10
3000	7.5
4000	6.5
6000	5.5
8000	5



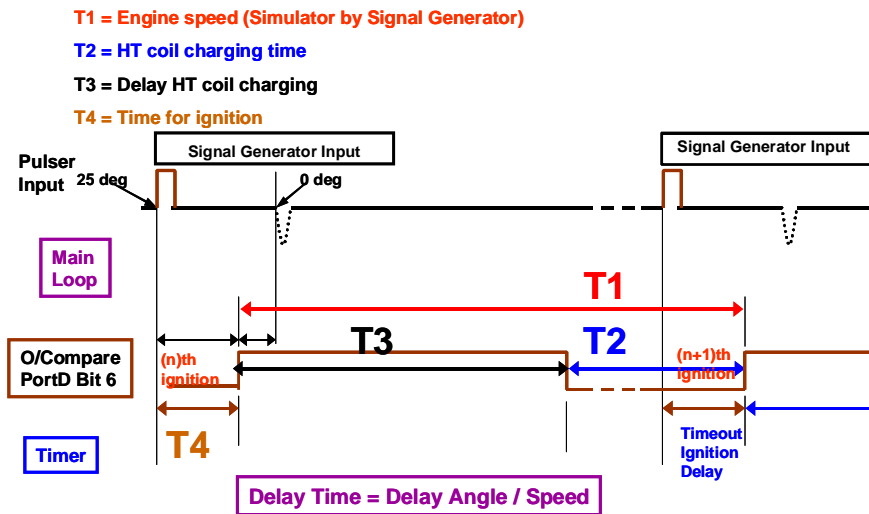
RPM	Advance Angle.
1000	3
1500	3
2500	17
3000	17
3500	25
8000	25



1.4.8 .Detail Waveform Description



1.4.9 Test waveform Description



Direct Current Ignition

1.4.10 Example

Example: 2400rpm

$$T1 = 60/2400 * \text{Crystal}/4 = 0.025 * 2000000 = 50000 \text{ cnt}$$

$$T2 = [\text{next cha_count} + (\text{next cha_spd} - \text{ARPM_Convert}) * \text{Cha_Factor}/256]$$

$$\text{next cha_count} = 7.5\text{ms}/(1/2\text{M}) = 15000$$

$$\text{next cha_spd}(3000\text{rpm}) = 3000 * 10 = 30000$$

$$\text{ARPM_Convert} = (\text{Crystal}/4) * 60 * 10 / T1 = 2\text{M} * 600 / 50000 = 24000$$

$$\text{Cha_factor} = 256 * (10\text{ms} - 7.5 \text{ms}) / (1/2\text{M}) / (3000\text{rpm} - 2000\text{rpm}) * 10 = 128$$

$$T2 = 15000 + ((30000 - 24000) * 128) / 256 = 18000 \text{ (9ms)}$$

$$T3 = T1 - T2 = 50000 - 18000 = 32000 \text{ (16ms)}$$

$T4 = \text{Step} + \text{Linear} \text{ (2400rpm)}$

$$\text{Step} = \text{DLY_STEP3} * \text{speed} = [(25-17) * (65536/360) * 60 / 2400 * (2\text{M})] / 65536 = 1110 \text{ cnt}$$

$$\begin{aligned} \text{Linear} &= (\text{Speed} - 2500\text{rpm}) * (2\text{M}) * (60/1500) * (17-3) * (65536/360) / \\ & \quad [(2\text{M}) * 60 * (1/1500 - 1/2500) * 65536] \\ &= 2\text{M} * [(60/2400 - 60/2500)] * (80000) * 14 * 182 / (32000) * 65536 \\ &= 194 \text{ cnt} \end{aligned}$$

$$T4 = 1110 + 194 = 1304 \text{ cnt}$$

$$\sim 650\text{us}$$

$$\begin{aligned} \text{Ignition angle} &= (T4 / T1) * 360 \text{ degree} \\ &= (1304 / 50000) * 360 = 9.38 \text{ degree} \end{aligned}$$

1.4.11 Test Table

Freq(Hz)	Speed(rpm)	T1(ms)	T2(ms)	T3(ms)	T4(ms)
15	900	66.67	**	**	**
16	960	62.5	20	42.5	3.81
20	1200	50	18	32	3.05
30	1800	33.33	12	21.33	1.64
40	2400	25	9	16	0.65
50	3000	20	7.5	12.5	0.44
60	3600	16.67	6.9	9.77	0.01
100	6000	10	5.5	4.5	0.01
140	8400	7.14	5	2.14	0.01
144	8640	6.94	Off	Off	Off

** Digital ignition is off and turns on analogue ignition, T2, T3 and T4 will depend on the actual mechanical angle of magneto.

Figure 1-8. Test Table

Figure 1-8 shows the testing result.

1.5 Extra Features

1.5.1 Hardware

Multi-ignition curves against analogue throttle input

- Remove the Q2 transistor to use the internal ADC channel 7 for the throttle analogue input potentiometer, in which the ADC (8bit) can recognize different voltage levels as an indication of the different throttle positions.

1.5.2 Firmware

- Adding the ADC conversion routine (see 1.5.3 ADC reference code) to convert the throttle input analogue signal to digital data.

Direct Current Ignition

1.5.3 ADC reference code

```

ADSCR      EQU      $003C      ;ANALOG TO DIGITAL STATUS AND CONTROL REGISTER
ADR        EQU      $003D      ;ANALOG TO DIGITAL DATA REGISTER
ADCLK      EQU      $003E      ;ANALOG TO DIGITAL INPUT CLOCK REGISTER
TEMP0      EQU      $90
TEMP1      EQU      $91
THOTTLE_POS EQU      $AA
SUM_ADC0   EQU      $B0
SUM_ADC1   EQU      $B1
CUR_ADC    EQU      $B2
AVR_ADC    EQU      $B3      ;AVERAGE ADC

;-----
;* ADC initialisation
;*
;* Date : 5/10/2000
;-----
INIT_ADC:
    LDA    #%00100111    ;INIT PORTB 7 AS ADC INPUT
    STA    ADSCR          ;START ADC CONVERSION WHILE BIT 5 SET
    LDA    #%00100000
    STA    ADCLK

INIT_ADC_EXIT:
    RTS

;-----
; Moving average
;
; Input      ADR
; Modify     TEMP0,1 SUM_ADC0,1 AVR_ADC
; Output     AVR_ADC= (SUM_ADC - SUM_ADC/4 + CUR_ADC)/4
;            THOTTLE= AVR_ADC
;
; Date : 5/11/2000
;-----
READ_ADC:
    LDA    ADR
    STA    CUR_ADC        ;UPD CUR_ADC

    LDA    SUM_ADC1       ;LOW BYTE
    STA    TEMP1
    LDA    SUM_ADC0       ;HI BYTE
    STA    TEMP0

    LSR    TEMP0          ;/2
    ROR    TEMP1

    LSR    TEMP0          ;/2
    ROR    TEMP1

    LDA    SUM_ADC1       ;TEMP0,1 = SUM_ADC/4
    SUB    TEMP1
    STA    SUM_ADC1
    LDA    SUM_ADC0
    SBC    TEMP0
    STA    SUM_ADC0       ;LESS SUM_ADC/4

    LDA    SUM_ADC1
    ADD    CUR_ADC
    STA    SUM_ADC1
    LDA    SUM_ADC0
    ADC    #0
    STA    SUM_ADC0

    LDA    SUM_ADC1
    STA    TEMP1
    LDA    SUM_ADC0
    STA    TEMP0

    LSRA           ;/2
    ROR    TEMP1
    LSRA           ;/2
    ROR    TEMP1

    LDA    TEMP1
    STA    AVR_ADC
    STA    THOTTLE_POS

    RTS

```

Freescale Semiconductor, Inc.

1.6 Further Information

1.6.1 Related Documents

MC68HC908JL3 Technical Data

AN2159/D Digital Direct Current Ignition System using HC08
Microcontrollers



Section 2. Glossary

A — See “accumulator (A).”

accumulator (A) — An 8-bit general-purpose register in the CPU08. The CPU08 uses the accumulator to hold operands and results of arithmetic and logic operations.

acquisition mode — A mode of PLL operation during startup before the PLL locks on a frequency. Also see “tracking mode.”

address bus — The set of wires that the CPU or DMA uses to read and write memory locations.

addressing mode — The way that the CPU determines the operand address for an instruction. The M68HC08 CPU has 16 addressing modes.

ALU — See “arithmetic logic unit (ALU).”

arithmetic logic unit (ALU) — The portion of the CPU that contains the logic circuitry to perform arithmetic, logic, and manipulation operations on operands.

asynchronous — Refers to logic circuits and operations that are not synchronized by a common reference signal.

baud rate — The total number of bits transmitted per unit of time.

BCD — See “binary-coded decimal (BCD).”

binary — Relating to the base 2 number system.

binary number system — The base 2 number system, having two digits, 0 and 1. Binary arithmetic is convenient in digital circuit design because digital circuits have two permissible voltage levels, low and high. The binary digits 0 and 1 can be interpreted to correspond to the two digital voltage levels.

binary-coded decimal (BCD) — A notation that uses 4-bit binary numbers to represent the 10 decimal digits and that retains the same positional structure of a decimal number. For example, 234 (decimal) = 0010 0011 0100 (BCD)

bit — A binary digit. A bit has a value of either logic 0 or logic 1.

branch instruction — An instruction that causes the CPU to continue processing at a memory location other than the next sequential address.

break module — A module in the M68HC08 Family. The break module allows software to halt program execution at a programmable point in order to enter a background routine.

breakpoint — A number written into the break address registers of the break module. When a number appears on the internal address bus that is the same as the number in the break address registers, the CPU executes the software interrupt instruction (SWI).

Glossary

- break interrupt** — A software interrupt caused by the appearance on the internal address bus of the same value that is written in the break address registers.
- bus** — A set of wires that transfers logic signals.
- bus clock** — The bus clock is derived from the CGMOUT output from the CGM. The bus clock frequency, f_{op} , is equal to the frequency of the oscillator output, CGMXCLK, divided by four.
- byte** — A set of eight bits.
- C** — The carry/borrow bit in the condition code register. The CPU08 sets the carry/borrow bit when an addition operation produces a carry out of bit 7 of the accumulator or when a subtraction operation requires a borrow. Some logical operations and data manipulation instructions also clear or set the carry/borrow bit (as in bit test and branch instructions and shifts and rotates).
- CCR** — See “condition code register.”
- central processor unit (CPU)** — The primary functioning unit of any computer system. The CPU controls the execution of instructions.
- CGM** — See “clock generator module (CGM).”
- clear** — To change a bit from logic 1 to logic 0; the opposite of set.
- clock** — A square wave signal used to synchronize events in a computer.
- clock generator module (CGM)** — A module in the M68HC08 Family. The CGM generates a base clock signal from which the system clocks are derived. The CGM may include a crystal oscillator circuit and or phase-locked loop (PLL) circuit.
- comparator** — A device that compares the magnitude of two inputs. A digital comparator defines the equality or relative differences between two binary numbers.
- computer operating properly module (COP)** — A counter module in the M68HC08 Family that resets the MCU if allowed to overflow.
- condition code register (CCR)** — An 8-bit register in the CPU08 that contains the interrupt mask bit and five bits that indicate the results of the instruction just executed.
- control bit** — One bit of a register manipulated by software to control the operation of the module.
- control unit** — One of two major units of the CPU. The control unit contains logic functions that synchronize the machine and direct various operations. The control unit decodes instructions and generates the internal control signals that perform the requested operations. The outputs of the control unit drive the execution unit, which contains the arithmetic logic unit (ALU), CPU registers, and bus interface.
- COP** — See “computer operating properly module (COP).”
- counter clock** — The input clock to the TIM counter. This clock is the output of the TIM prescaler.
- CPU** — See “central processor unit (CPU).”
- CPU08** — The central processor unit of the M68HC08 Family.
- CPU clock** — The CPU clock is derived from the CGMOUT output from the CGM. The CPU clock frequency is equal to the frequency of the oscillator output, CGMXCLK, divided by four.

CPU cycles — A CPU cycle is one period of the internal bus clock, normally derived by dividing a crystal oscillator source by two or more so the high and low times will be equal. The length of time required to execute an instruction is measured in CPU clock cycles.

CPU registers — Memory locations that are wired directly into the CPU logic instead of being part of the addressable memory map. The CPU always has direct access to the information in these registers. The CPU registers in an M68HC08 are:

- A (8-bit accumulator)
- H:X (16-bit index register)
- SP (16-bit stack pointer)
- PC (16-bit program counter)
- CCR (condition code register containing the V, H, I, N, Z, and C bits)

CSIC — customer-specified integrated circuit

cycle time — The period of the operating frequency: $t_{CYC} = 1/f_{OP}$.

decimal number system — Base 10 numbering system that uses the digits zero through nine.

direct memory access module (DMA) — A M68HC08 Family module that can perform data transfers between any two CPU-addressable locations without CPU intervention. For transmitting or receiving blocks of data to or from peripherals, DMA transfers are faster and more code-efficient than CPU interrupts.

DMA — See “direct memory access module (DMA).”

DMA service request — A signal from a peripheral to the DMA module that enables the DMA module to transfer data.

duty cycle — A ratio of the amount of time the signal is on versus the time it is off. Duty cycle is usually represented by a percentage.

EEPROM — Electrically erasable, programmable, read-only memory. A nonvolatile type of memory that can be electrically reprogrammed.

EPROM — Erasable, programmable, read-only memory. A nonvolatile type of memory that can be erased by exposure to an ultraviolet light source and then reprogrammed.

exception — An event such as an interrupt or a reset that stops the sequential execution of the instructions in the main program.

external interrupt module (IRQ) — A module in the M68HC08 Family with both dedicated external interrupt pins and port pins that can be enabled as interrupt pins.

fetch — To copy data from a memory location into the accumulator.

firmware — Instructions and data programmed into nonvolatile memory.

free-running counter — A device that counts from zero to a predetermined number, then rolls over to zero and begins counting again.

full-duplex transmission — Communication on a channel in which data can be sent and received simultaneously.

Glossary

H — The upper byte of the 16-bit index register (H:X) in the CPU08.

H — The half-carry bit in the condition code register of the CPU08. This bit indicates a carry from the low-order four bits of the accumulator value to the high-order four bits. The half-carry bit is required for binary-coded decimal arithmetic operations. The decimal adjust accumulator (DAA) instruction uses the state of the H and C bits to determine the appropriate correction factor.

hexadecimal — Base 16 numbering system that uses the digits 0 through 9 and the letters A through F.

high byte — The most significant eight bits of a word.

illegal address — An address not within the memory map

illegal opcode — A nonexistent opcode.

I — The interrupt mask bit in the condition code register of the CPU08. When I is set, all interrupts are disabled.

index register (H:X) — A 16-bit register in the CPU08. The upper byte of H:X is called H. The lower byte is called X. In the indexed addressing modes, the CPU uses the contents of H:X to determine the effective address of the operand. H:X can also serve as a temporary data storage location.

input/output (I/O) — Input/output interfaces between a computer system and the external world. A CPU reads an input to sense the level of an external signal and writes to an output to change the level on an external signal.

instructions — Operations that a CPU can perform. Instructions are expressed by programmers as assembly language mnemonics. A CPU interprets an opcode and its associated operand(s) and instruction.

interrupt — A temporary break in the sequential execution of a program to respond to signals from peripheral devices by executing a subroutine.

interrupt request — A signal from a peripheral to the CPU intended to cause the CPU to execute a subroutine.

I/O — See “input/output (I/O).”

IRQ — See “external interrupt module (IRQ).”

jitter — Short-term signal instability.

latch — A circuit that retains the voltage level (logic 1 or logic 0) written to it for as long as power is applied to the circuit.

latency — The time lag between instruction completion and data movement.

least significant bit (LSB) — The rightmost digit of a binary number.

logic 1 — A voltage level approximately equal to the input power voltage (V_{DD}).

logic 0 — A voltage level approximately equal to the ground voltage (V_{SS}).

low byte — The least significant eight bits of a word.

low voltage inhibit module (LVI) — A module that monitors power supply voltage.

LVI — See “low voltage inhibit module (LVI).”

M68HC08 — A Motorola family of 8-bit MCUs.

mark/space — The logic 1/logic 0 convention used in formatting data in serial communication.

mask — 1. A logic circuit that forces a bit or group of bits to a desired state. 2. A photomask used in integrated circuit fabrication to transfer an image onto silicon.

mask option — A optional microcontroller feature that the customer chooses to enable or disable.

mask option register (MOR) — An EPROM location containing bits that enable or disable certain MCU features.

MCU — Microcontroller unit. See “microcontroller.”

memory location — Each M68HC08 memory location holds one byte of data and has a unique address. To store information in a memory location, the CPU places the address of the location on the address bus, the data information on the data bus, and asserts the write signal. To read information from a memory location, the CPU places the address of the location on the address bus and asserts the read signal. In response to the read signal, the selected memory location places its data onto the data bus.

memory map — A pictorial representation of all memory locations in a computer system.

microcontroller — Microcontroller unit (MCU). A complete computer system, including a CPU, memory, a clock oscillator, and input/output (I/O) on a single integrated circuit.

modulo counter — A counter that can be programmed to count to any number from zero to its maximum possible modulus.

monitor ROM — A section of ROM that can execute commands from a host computer for testing purposes.

MOR — See “mask option register (MOR).”

most significant bit (MSB) — The leftmost digit of a binary number.

multiplexer — A device that can select one of a number of inputs and pass the logic level of that input on to the output.

N — The negative bit in the condition code register of the CPU08. The CPU sets the negative bit when an arithmetic operation, logical operation, or data manipulation produces a negative result.

nibble — A set of four bits (half of a byte).

object code — The output from an assembler or compiler that is itself executable machine code, or is suitable for processing to produce executable machine code.

opcode — A binary code that instructs the CPU to perform an operation.

open-drain — An output that has no pullup transistor. An external pullup device can be connected to the power supply to provide the logic 1 output voltage.

operand — Data on which an operation is performed. Usually a statement consists of an operator and an operand. For example, the operator may be an add instruction, and the operand may be the quantity to be added.

oscillator — A circuit that produces a constant frequency square wave that is used by the computer as a timing and sequencing reference.

Glossary

- OTPROM** — One-time programmable read-only memory. A nonvolatile type of memory that cannot be reprogrammed.
- overflow** — A quantity that is too large to be contained in one byte or one word.
- page zero** — The first 256 bytes of memory (addresses \$0000–\$00FF).
- parity** — An error-checking scheme that counts the number of logic 1s in each byte transmitted. In a system that uses odd parity, every byte is expected to have an odd number of logic 1s. In an even parity system, every byte should have an even number of logic 1s. In the transmitter, a parity generator appends an extra bit to each byte to make the number of logic 1s odd for odd parity or even for even parity. A parity checker in the receiver counts the number of logic 1s in each byte. The parity checker generates an error signal if it finds a byte with an incorrect number of logic 1s.
- PC** — See “program counter (PC).”
- peripheral** — A circuit not under direct CPU control.
- phase-locked loop (PLL)** — A oscillator circuit in which the frequency of the oscillator is synchronized to a reference signal.
- PLL** — See “phase-locked loop (PLL).”
- pointer** — Pointer register. An index register is sometimes called a pointer register because its contents are used in the calculation of the address of an operand, and therefore points to the operand.
- polarity** — The two opposite logic levels, logic 1 and logic 0, which correspond to two different voltage levels, V_{DD} and V_{SS} .
- polling** — Periodically reading a status bit to monitor the condition of a peripheral device.
- port** — A set of wires for communicating with off-chip devices.
- prescaler** — A circuit that generates an output signal related to the input signal by a fractional scale factor such as 1/2, 1/8, 1/10 etc.
- program** — A set of computer instructions that cause a computer to perform a desired operation or operations.
- program counter (PC)** — A 16-bit register in the CPU08. The PC register holds the address of the next instruction or operand that the CPU will use.
- pull** — An instruction that copies into the accumulator the contents of a stack RAM location. The stack RAM address is in the stack pointer.
- pullup** — A transistor in the output of a logic gate that connects the output to the logic 1 voltage of the power supply.
- pulse-width** — The amount of time a signal is on as opposed to being in its off state.
- pulse-width modulation (PWM)** — Controlled variation (modulation) of the pulse width of a signal with a constant frequency.
- push** — An instruction that copies the contents of the accumulator to the stack RAM. The stack RAM address is in the stack pointer.
- PWM period** — The time required for one complete cycle of a PWM waveform.

RAM — Random access memory. All RAM locations can be read or written by the CPU. The contents of a RAM memory location remain valid until the CPU writes a different value or until power is turned off.

RC circuit — A circuit consisting of capacitors and resistors having a defined time constant.

read — To copy the contents of a memory location to the accumulator.

register — A circuit that stores a group of bits.

reserved memory location — A memory location that is used only in special factory test modes. Writing to a reserved location has no effect. Reading a reserved location returns an unpredictable value.

reset — To force a device to a known condition.

ROM — Read-only memory. A type of memory that can be read but cannot be changed (written). The contents of ROM must be specified before manufacturing the MCU.

SCI — See “serial communication interface module (SCI).”

serial — Pertaining to sequential transmission over a single line.

serial communications interface module (SCI) — A module in the M68HC08 Family that supports asynchronous communication.

serial peripheral interface module (SPI) — A module in the M68HC08 Family that supports synchronous communication.

set — To change a bit from logic 0 to logic 1; opposite of clear.

shift register — A chain of circuits that can retain the logic levels (logic 1 or logic 0) written to them and that can shift the logic levels to the right or left through adjacent circuits in the chain.

signed — A binary number notation that accommodates both positive and negative numbers. The most significant bit is used to indicate whether the number is positive or negative, normally logic 0 for positive and logic 1 for negative. The other seven bits indicate the magnitude of the number.

software — Instructions and data that control the operation of a microcontroller.

software interrupt (SWI) — An instruction that causes an interrupt and its associated vector fetch.

SPI — See “serial peripheral interface module (SPI).”

stack — A portion of RAM reserved for storage of CPU register contents and subroutine return addresses.

stack pointer (SP) — A 16-bit register in the CPU08 containing the address of the next available storage location on the stack.

start bit — A bit that signals the beginning of an asynchronous serial transmission.

status bit — A register bit that indicates the condition of a device.

stop bit — A bit that signals the end of an asynchronous serial transmission.

Glossary

subroutine — A sequence of instructions to be used more than once in the course of a program. The last instruction in a subroutine is a return from subroutine (RTS) instruction. At each place in the main program where the subroutine instructions are needed, a jump or branch to subroutine (JSR or BSR) instruction is used to call the subroutine. The CPU leaves the flow of the main program to execute the instructions in the subroutine. When the RTS instruction is executed, the CPU returns to the main program where it left off.

synchronous — Refers to logic circuits and operations that are synchronized by a common reference signal.

TIM — See “timer interface module (TIM).”

timer interface module (TIM) — A module used to relate events in a system to a point in time.

timer — A module used to relate events in a system to a point in time.

toggle — To change the state of an output from a logic 0 to a logic 1 or from a logic 1 to a logic 0.

tracking mode — Mode of low-jitter PLL operation during which the PLL is locked on a frequency. Also see “acquisition mode.”

two's complement — A means of performing binary subtraction using addition techniques. The most significant bit of a two's complement number indicates the sign of the number (1 indicates negative). The two's complement negative of a number is obtained by inverting each bit in the number and then adding 1 to the result.

unbuffered — Utilizes only one register for data; new data overwrites current data.

unimplemented memory location — A memory location that is not used. Writing to an unimplemented location has no effect. Reading an unimplemented location returns an unpredictable value. Executing an opcode at an unimplemented location causes an illegal address reset.

V — The overflow bit in the condition code register of the CPU08. The CPU08 sets the V bit when a two's complement overflow occurs. The signed branch instructions BGT, BGE, BLE, and BLT use the overflow bit.

variable — A value that changes during the course of program execution.

VCO — See “voltage-controlled oscillator.”

vector — A memory location that contains the address of the beginning of a subroutine written to service an interrupt or reset.

voltage-controlled oscillator (VCO) — A circuit that produces an oscillating output signal of a frequency that is controlled by a dc voltage applied to a control input.

waveform — A graphical representation in which the amplitude of a wave is plotted against time.

wired-OR — Connection of circuit outputs so that if any output is high, the connection point is high.

word — A set of two bytes (16 bits).

write — The transfer of a byte of data from the CPU to a memory location.

X — The lower byte of the index register (H:X) in the CPU08.

Z — The zero bit in the condition code register of the CPU08. The CPU08 sets the zero bit when an arithmetic operation, logical operation, or data manipulation produces a result of \$00.



Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

**For More Information On This Product,
Go to: www.freescale.com**

HOW TO REACH US:**USA/EUROPE/LOCATIONS NOT LISTED:**

Motorola Literature Distribution;
P.O. Box 5405, Denver, Colorado 80217
1-303-675-2140 or 1-800-441-2447

JAPAN:

Motorola Japan Ltd.; SPS, Technical Information Center,
3-20-1, Minami-Azabu Minato-ku, Tokyo 106-8573 Japan
81-3-3440-3569

ASIA/PACIFIC:

Motorola Semiconductors H.K. Ltd.;
Silicon Harbour Centre, 2 Dai King Street,
Tai Po Industrial Estate, Tai Po, N.T., Hong Kong
852-26668334

TECHNICAL INFORMATION CENTER:

1-800-521-6274

HOME PAGE:

<http://motorola.com/semiconductors>

Information in this document is provided solely to enable system and software implementers to use Motorola products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Motorola data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part.



Motorola and the Stylized M Logo are registered in the U.S. Patent and Trademark Office. digital dna is a trademark of Motorola, Inc. All other product or service names are the property of their respective owners. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

© Motorola, Inc. 2003