DAVICOM Semiconductor, Inc.

DM120C16E

Mixed Signal 8-bit MCU for EPD Application

DATA SHEET

Preliminary

Version: DM120C16E-11-MCO-DS-P01

November 28, 2014

Content

1 General Description

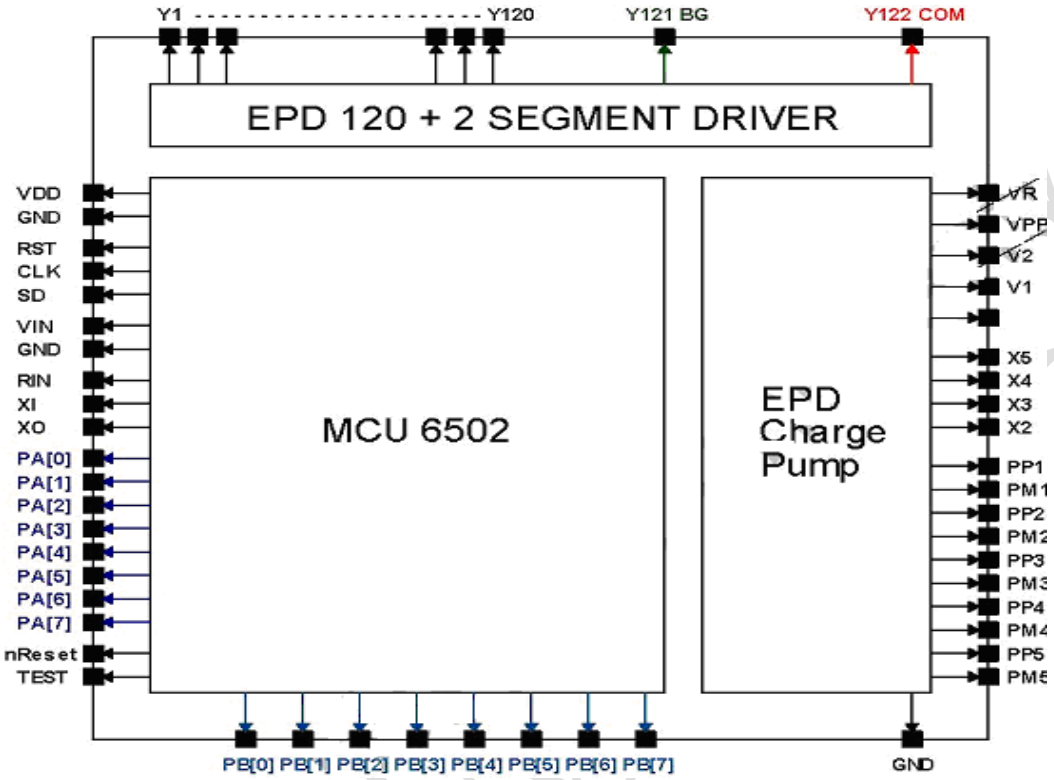
The DM120C16E is embedded 8-bit 6502 MCU and consist of 120 segments and Common/Background electrode EPD driver. The chip provides ISO-7816 contact interface for Smart card application.

2 Features

- **Working voltage：2V ~ 5.5V**
- **CMOS low power consumption**
- **Common electrode have 50uA sink/source current driving capability**
- **Background electrode have 50uA sink/source current driving capability**
- **120 segments drive up to 10uA sink/source current capability**
- **4 output modes are available in Hi-V channels (-5V , 15V , 30V , Hi-impedance)**
- **Built-in DC-DC charge pump circuit for EPD application**
- **Power management mode ("Sleep mode" operating current ≦ 50nA @ 3V, typical)**
- **Power-on reset (POR)**
- **Low voltage reset (LVR)**
- **8-bit A/D converter for detecting working voltage or temperature**
- **2(KV) ESD protection (smart card interface VDD, GND, RST, CLK, DATA up to 4KV)**

Internal RC Oscillator	3.5MHz
CPU architecture	8-bit 6502
Program ROM	16K * 8-bits
SRAM	1024 * 8-bits
Special Function Register (SFR)	32 * 8-bits
STACK RAM	256 * 8-bits
Interrupt Source	System Reset, ISO-7816 RST, ISO-7816 I/O,
	Timer0, Timer1, GPIO [PA], GPIO [PB] and RTC
Selectable main clock	3.5Mhz / 1.75Mhz / 875Khz / 437Khz
Two system clock	Internal main clock & External 32Khz
GPIO	8 channels Port A and 8 channels Port B
Serial protocol	ISO-7816, SPI, UART
Power consumption	Sleep mode ≦ 50nA (VDD=3V)
	EPD operating mode ≦400uA (VDD=3V @ 3.5Mhz)
EPD driver pin	Segment： 120
	Common： 1
	Background： 1
	Driving voltage：-4V, 18V, 36V
Built-in two Timer	16-bit Timer0 / 16-bit Timer1
Built-in RTC (Real time clock)	Hour / minute / second
Power Saving mode	Sleep / Halt

3 Block Diagram



4 PAD Diagram

5 Pin Description

Symbol	I/O	Description
VDD	Power	Positive power input for digital system
VBAT	Power	For external battery power supply
GND	Power	Negative power input for digital system
AGND	Power	Negative power input for analog system
Feqtest	O	Internal circuit short to GND
RIN	I	User has to connect a 0.01uF cap. to V1D5 pad
nReset	I	System reset pin
nTest	I	User has to connecting this pin to VDD
Xi	I	32.768KHz crystal input terminal
Xo	O	32.768KHz crystal output terminal
GPIO PortA[7~0]	I/O	General purpose input / output PortA[7~0]
GPIO PortB[0] / Buzzer-	I/O	General purpose input / output PortB[0] or buzzer output -
GPIO PortB[1] / Buzzer+	I/O	General purpose input / output PortB[1] or buzzer output +
GPIO PortB[7~2]	I/O	General purpose input / output Port B[7:2]
RST	I/O	ISO 7816 Reset pin. *Please refer to ISO 7816 standard*
CLK	I/O	ISO 7816 Clock pin
SD	I/O	ISO 7816 Data pin
Y[122] / COM	O	EPD Common terminal
Y[121] / BG	O	EPD Background terminal
Y[120~1] / SEG	O	EPD segment terminals
VPP	I/O	EPD High Voltage
V1D5	-	1.5V output
X5	-	Charge pump X5
X4	-	Charge pump X4
X3	-	Charge pump X3
X2	-	Charge pump X2
PP1	-	Charge pump C1 +
PM1	-	Charge pump C1 -
PP2	-	Charge pump C2 +
PM2	-	Charge pump C2 -
PP3	-	Charge pump C3 +
PM3	-	Charge pump C3 -
PP4	-	Charge pump C4 +
PM4	-	Charge pump C4 -
PP5	-	Charge pump C5 +
PM5	-	Charge pump C5 -

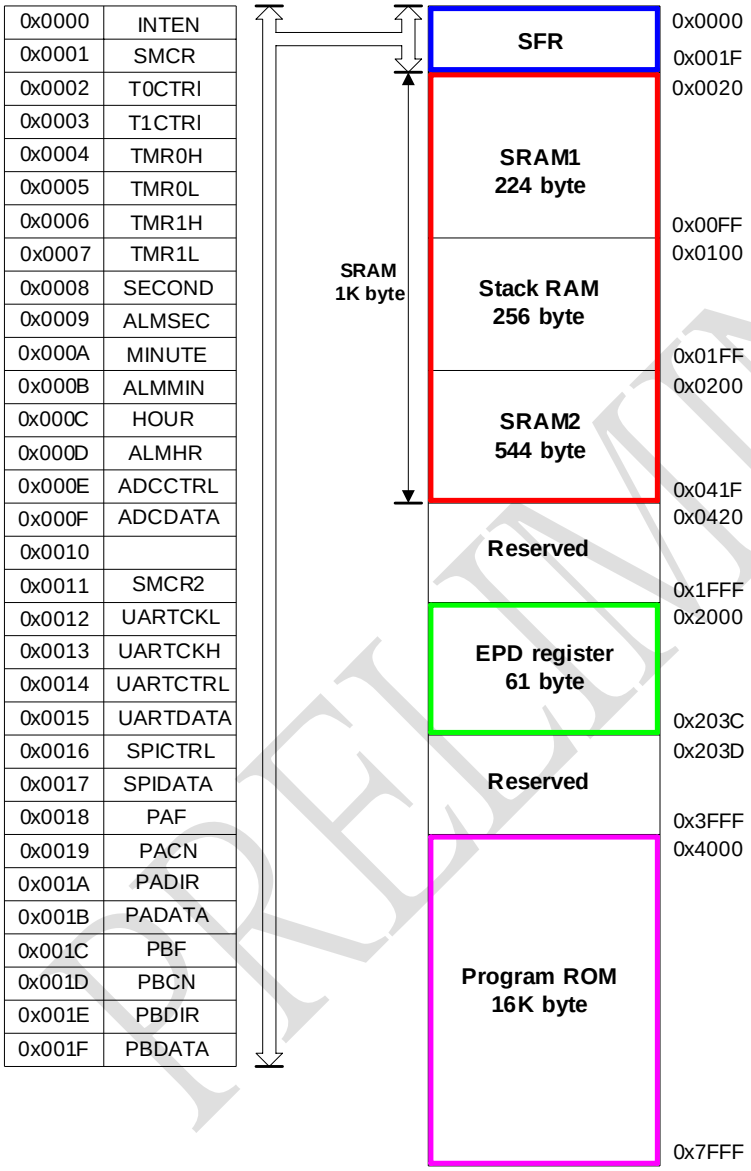
Note:
1. We suggest user add 0.01uF capacitor on “RIN” & “V1D5” pin for against noise at power saving “Sleep mode”
2. If IC powered with (3V) battery that PAD VBAT & VDD shouldn’t short while using ISO 7816.

6 Memory Mapping

Special Function Register：32 Bytes
User RAM Size：768 Bytes
Stack RAM Size：256 Bytes
EPD RAM Size：61 Bytes
ROM Size：16K Bytes

Special function register

0x0000	INTEN
0x0001	SMCR
0x0002	T0CTRI
0x0003	T1CTRI
0x0004	TMR0H
0x0005	TMR0L
0x0006	TMR1H
0x0007	TMR1L
0x0008	SECOND
0x0009	ALMSEC
0x000A	MINUTE
0x000B	ALMMIN
0x000C	HOUR
0x000D	ALMHR
0x000E	ADCCTRL
0x000F	ADCDATA
0x0010	
0x0011	SMCR2
0x0012	UARTCKL
0x0013	UARTCKH
0x0014	UARTCTRL
0x0015	UARTDATA
0x0016	SPICTRL
0x0017	SPIDATA
0x0018	PAF
0x0019	PACN
0x001A	PADIR
0x001B	PADATA
0x001C	PBF
0x001D	PBCN
0x001E	PBDIR
0x001F	PBDATA

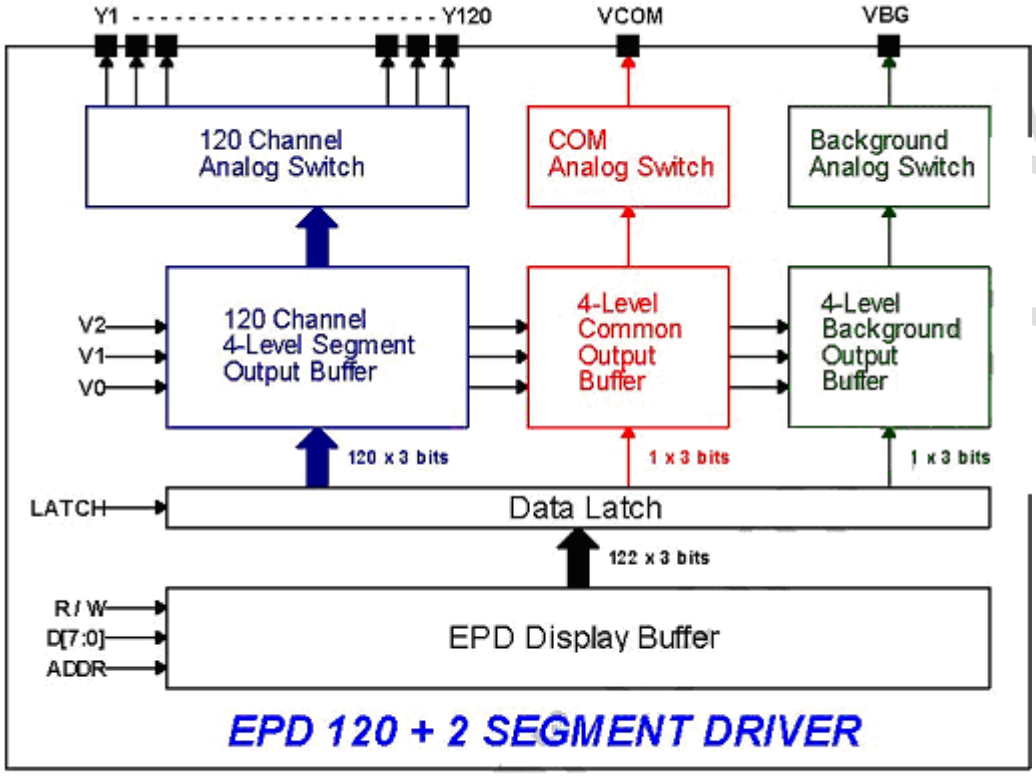


	b7	b6 b5 b4	b3	b2 b1 b0
0x2000	--	Y[2]	--	Y[1]
0x2001	--	Y[4]	--	Y[3]
⋮		⋮		
0x203B	--	Y[120]	--	Y[119]
0x203C	--	Y[bg]	--	Y[com]

7 EPD Driver Description

7.1 EPD Block Diagram

The chip consists of 120 channels EPD driver including common / background / segment.



7.2 EPD Control Register

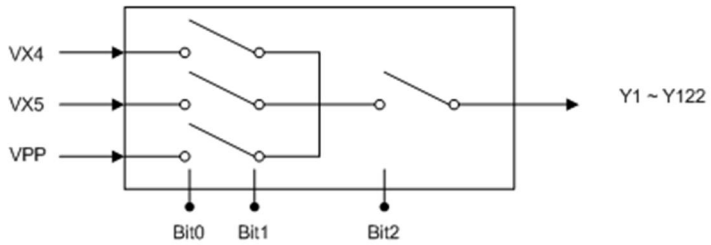
Each Hi-V channel are controlled as the follow register:

The EPD display control register are located at MPU data memory from address 0x2000 to 0x203C. Data will output after LATCH signal is going from High to Low.
*D3/bit3 & D7/bit3 are dummy bits

Address	0x2000	0x2001	0x2002	...	0x203B	0x203C	0x203D	0x203E	0x203F
D0/Bit0	Y1.0	Y3.0	Y5.0	...	Y119.0	VCOM.0	--	--	--
D1/Bit1	Y1.1	Y3.1	Y5.1	...	Y119.1	VCOM.1	--	--	--
D2/Bit2	Y1.2	Y3.2	Y5.2	...	Y119.2	VCOM.2	--	--	--
D3/Bit3	--	--	--	...	--	--	--	--	--
D4/Bit0	Y2.0	Y4.0	Y6.0	...	Y120.0	VBG.0	--	--	--
D5/Bit1	Y2.1	Y4.1	Y6.1	...	Y120.1	VBG.1	--	--	--
D6/Bit2	Y2.2	Y4.2	Y6.2	...	Y120.2	VBG.2	--	--	--
D7/Bit3	--	--	--	...	--	--	--	--	--

Hi-V channel control register

Circuit diagram for each output driver:
All of the output drivers are the same configuration. Include Y1…Y120, VCOM and VBG.
VX5=Half VPP, VX4=negative voltage (-5V)



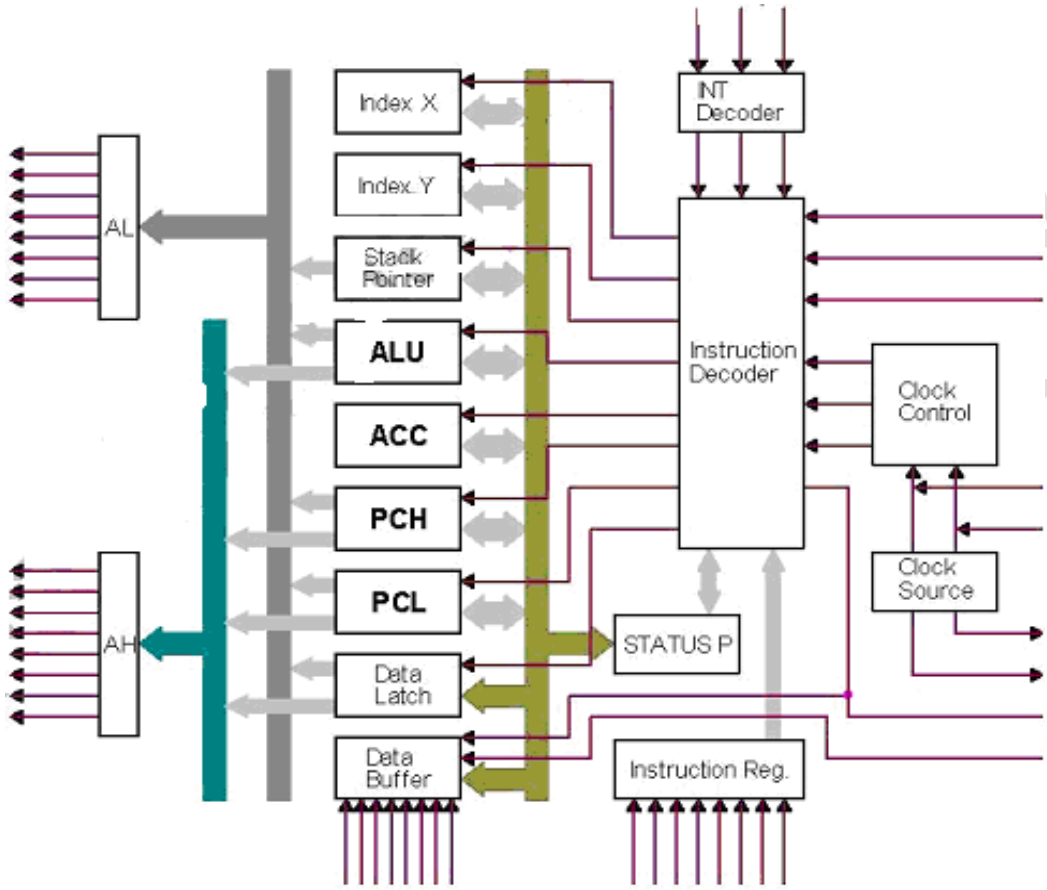
Bit1 and Bit0 select the source voltage, Bit2 controls the output switch.

Bit2/Bit1/Bit0	Voltage	Output Switch
000	VX4	Off
001	VX5	Off
010	VX5	Off
011	VPP	Off
100	VX4	On
101	VX5	On
110	VX5	On
111	VPP	On

Output Status Truth Table

8 CPU & Register

8.1 CPU Block Diagram



8.2 Basic Mode Status Register (BMSR) – 01

For cost & efficiency optimum application. Here designed in 8-Bit CPU.

8.2.1 CPU Description

- 8-Bits Data bus
- 16-Bits Address bus could access 64K bytes memory space.
- 95 instructions
- 12 addressing modes
- 16-Bits program counter
- 8-Bits Registers, including Accumulator, Index register, Processor status register
- Faster instruction cycle time

8.3 PHY ID Identifier Register #1 (PHYID1) - 02

There are several registers in the following items. These registers are dedicated to CPU. For example “ALU” is accessing procedure.

8.3.1 Accumulator (A)

The accumulator is an 8-bit register used for holding one of the operands to do exchange, arithmetic or logical operation.

8.3.2 Index Register (X & Y)

Internality build in two index register (X & Y). These two index-registers can use as counting program steps or to provide an index value for generating an effective address. When executing an instruction that specifies indexed addressing, the CPU fetches the operation code and the base address, and then modifies the address by adding the index register to it prior performing operation.

8.3.3 Stack Pointer Register (SP)

The stack pointer is an 8-bit register which stores the return addresses while serving subroutine or interrupt procedure. (SP) is automatically incremented or decremented under control of the microprocessor to perform stack manipulations under parts of either subroutine or interrupts (NMIB and IRQB). (SP) allows simple implementation of nested subroutines and multiple level interrupts. The stack pointer is initialized by software definition.

8.3.4 Program Counter (PC)

The 16-bit program counter register provides the address that step the microprocessor through sequential program instructions. Microprocessor will fetch current data from (PC) for next execution instruction. After microprocessor fetches the instruction correctly then the (PC) will be incremented “1” automatically.

8.3.5 Processor Status Register (P)

The 8-bit status flag register contains seven status flags. Some of these flags are controlled by software and others are both controlled by software and the CPU. With software these status flags are tested within conditional branch instructions.

Bit	Symbol	Description
7	N	Negative，high = true
6	V	Overflow，high = true
5	---	
4	---	
3	D	Decimal mode，high = true
2	I	Interrupt Inhibit，high = true
1	Z	Zero，high = true
0	C	Carry，high = true

8.4 Special Function Register (SFR)

The DM120C16E design in some special function registers located in zero-page address. Program access SFR to set function like Timer, A/D, System clock…etc. User can refer to correspondence sections and getting more detail.

8.5 Addressing Modes

There are twelve of addressing modes for application. Take reference in the following descriptions that user can chose the most efficiency mode in programming.

8.5.1 Immediate Addressing

In immediate addressing, For example：(LDA #3FH) the first “LDA” is OP code and the second “#3FH” of the instruction content is operand, totally occupied two byte in memory space.

Byte	2	1	0
Instruction symbol		Operand	OP code
Operand			Operand

8.5.2 Implied Addressing

In the implied addressing mode, the address contains the operand that is implicit in the operation code of the instruction. It means this mode only occupied one byte. For example：(CLC、DEX、TAX…etc.)

Byte	2	1	0
Instruction symbol			OP code
Operand			Implied

8.5.3 Accumulator Addressing

This mode is part of implied addressing and four instructions are included. The form of addressing is represented as one-byte instruction which implies an operation in the accumulator. For example：(ASL A、LSR A、ROR A、ROL A)

Byte	2	1	0
Instruction symbol			OP code
Operand			Accumulator

8.5.4 Relative Addressing

Relative addressing is only with branch instructions and established by a destination for the conditional branch. For example：(BCC、BCS、BEQ、BNE.…etc.)

The second byte of the instruction becomes the operand which is an “offset” added to the contents of the lower eight bits of the program counter when the counter is setting to next instruction. The range of the offset is –128 to +127 bytes in the next instruction.

Byte	2	1	0
Instruction symbol		offset	OP code
		PCH	PCL
			offset
New PC value		effective address	

8.5.5 Absolute Addressing

In absolute addressing, the second byte of the instruction specifies the eight low order bits of the effective address while the third byte specifies the eight high order bits. Thus the absolute addressing mode allows accessing to the entire 64K bytes of addressable memory. For example：(LDA $2000H)

Byte	2	1	0
Instruction symbol	ADH	ADL	OP code
Operand address		ADH	ADL

8.5.6 Absolute Indexed Addressing

(X, Y indexing) – This form of addressing is applied to conjunct X and Y index register and also be referred to “absolute, X” and “absolute, Y”. The effective address is formed by adding the contents of X or Y to the basic address which is in the second and the third bytes of the instruction. This mode applies the index register to contain the index or countable value adding to the basic address. The index in this mode that allows any location to be reference. The purpose is reducing coding and execution time.

Byte	2	1	0
Instruction symbol	ADH	ADL	OP code
Absolute base address		ADH	ADL
+index			X or Y
Operand address		effective address	

8.5.7 Absolute Indexed Indirect Addressing

The X Index Register is added to the second and third byes of the instruction to form an address to a pointer. At this mode CPU will fetch second and third byte then add index X to form an effective address.

Byte	2	1	0
Instruction symbol	ADH	ADL	OP code
Indirect base address		ADH	ADL
+index			X
Indirect address		effective address	
New PC value		Indirect address	

8.5.8 Absolute Indirect Addressing

The second byte of the instruction contains the low order eight bits of a memory location. The high order eight bits of that memory location are contained in the third byte of the instruction. The contents of the fully specified memory location are the low order byte of the effective address. The next memory location contains the high order byte of the effective address which is loaded into the sixteen bits of the program counter. (JMP (ABS) only)

Byte	2	1	0
Instruction symbol	ADH	ADL	OP code
Indirect address		ADH	ADL
New PC value		Indirect address	

8.5.9 Zero Page Addressing

Here is one of absolute addressing mode. The CPU will fetch most top 256 bytes of memory spaces. This addressing in Hexadecimal represent as “$0000H~$00FFH”. All the high bytes are the same equal to “00”. So, it define as Zero page addressing. For example：(LDA $2AH), it occupied two byte the OP code “LDA” combine with “$2AH” address.

Byte	2	1	0
Instruction symbol		ZP	OP code
Operand address			ZP

8.5.10 Zero Page Indexed Addressing

(X, Y indexing) – This form of addressing is applied to the index register and is referred to as “Zero Page, X” or “Zero Page, Y”. The effective address is calculated by adding the second byte to the contents of the index register. Since this is a form of “Zero Page” addressing, the content of the second byte references a location in page zero. Additionally, due to the “Zero Page” addressing nature of this mode, no carry is added to the high order eight bits of memory and crossing of page boundaries does not occur.

Byte	2	1	0
Instruction symbol		ZP	OP code
Base address			ZP
+index			X or Y
Operand address		0	effective address

8.5.11 Zero Page Indexed Indirect Addressing

In this addressing mode the second byte of the instruction is added to the contents of the X index register, discarding the carry. The result of this addition points to a memory location on zero page whose contains are the low eight bits of the effective address. The next memory location in zero page contains the eight bits of the effective address. Both memory locations specifying the high and low order bytes of the effective address must be in zero page.

Byte	2	1	0
Instruction symbol		ZP	OP code
Base address			ZP
+index			X
Indirect address		0	address
Operand address		Indirect address	

8.5.12 Zero Page Indirect Addressing

This mode is similar to 8.5.8 absolute indirect addressing, but second byte operand only located in zero page. For example：(JMP $3AH)

Byte	2	1	0
Instruction symbol		ZP	OP code
Indirect address		0	ZP
Operand address		Indirect address	

8.6 Index Table of Operation Codes

MSB \ LSB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	BRK s	ORA (zp,x)			TSB zp	ORA zp	ASL zp		PHP s	ORA #	ASL A			ORA a	ASL a	
1	BPL r	ORA (zp),y			TRB zp	ORA zp,x	ASL zp,x		CLC i	ORA a,y	INC A			ORA a,x	ASL a,x	
2	JSR a	AND (zp,x)			BIT zp	AND zp	ROL zp		PLP s	AND #	ROL A		BIT a	AND a	ROL a	
3	BMI r	AND (zp),y			BIT zp,x	AND zp,x	ROL zp,x		SEC i	AND a,y	DEC A		BIT a,x	AND a,x	ROL a,x	
4	RTI s	EOR (zp,x)				EOR zp	LSR zp		PHA s	EOR #	LSR A		JMP a	EOR a	LSR a	
5	BVC r	EOR (zp),y				EOR zp,x	LSR zp,x		CLI i	EOR a,y	PHY s			EOR a,x	LSR a,x	
6	RTS s	ADC (zp,x)			STZ zp	ADC zp	ROR zp		PLA s	ADC #	ROR A		JMP (a)	ADC a	ROR a	
7	BVS r	ADC (zp),y			STZ zp,x	ADC zp,x	ROR zp,x		SEI i	ADC a,y	PLY s		JMP (a,x)	ADC a,x	ROR a,x	
8	BRA r	STA (zp,x)			STY zp	STA zp	STZ zp		DEY i	BIT #	TXA i		STY a	STA a	STX a	
9	BCC r	STA (zp),y			STY zp,x	STA zp,x	STZ zp,x		TYA i	STA a,y	TXS i		STZ a	STA a,x	STZ a,x	
A	LDY #	LDA (zp,x)			LDY zp	LDA zp	LDX zp		TAY i	LDA #	TAX i		LDY a	LDA a	LDX a	
B	BCS r	LDA (zp),y			LDY zp,x	LDA zp,x	LDX zp,x		CLV i	LDA a,y	TSX i		LDY a,x	LDA a,x	LDX a,x	
C	CPY #	CMP (zp,x)			CPY zp	CMP zp	DEC zp		INY i	CMP #	DEX i		CPY a	CMP a	DEC a	
D	BNE r	CMP (zp),y				CMP zp,x	DEC zp,x		CLD i	CMP a,y	PHX s			CMP a,x	DEC a,x	
E	CPX #	SBC (zp,x)			CPX zp	SBC zp	INC zp		INX i	SBC #	NOP i		CPX a	SBC a	INC a	
F	BEQ r	SBC (zp),y				SBC zp,x	INC zp,x		SED i	SBC a,y	PLX s			SBC a,x	INC a,x	
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**

Note!
a：absolute addressing
A：accumulator addressing
r：conditional branch addressing
s：stack ram push & pop
#：Immediately addressing
zp：Zero page addressing
i：Implied addressing

8.7 Operation Codes & Status Register

Instructions	Operation # immediate data ~ NOT ^ AND v OR ⊕ Exclusive OR	Addressing modes																Status register					
		a	(a,x)	a,x	a,y	(a)	A	#	i	r	s	zp	(zp,x)	zp,x	zp,y	(zp)	(zp),y	N	V	D	I	Z	C
ADC	A+M+C→A	6D		7D	79			69				65	61	75		72	71	N	V	-	-	Z	C
AND	A^M→A	2D		3D	39			29				25	21	35		32	31	N	-	-	-	Z	-
ASL	C←MSB~LSB←0	0E		1E			0A					06		16				N	-	-	-	Z	C
BCC	Branch C=0									90								-	-	-	-	-	-
BCS	Branch C=1									B0								-	-	-	-	-	-
BEQ	Branch Z=1									F0								-	-	-	-	-	-
BIT	A^M	2C		3C				89				24		34				M_7	M_6	-	-	Z	-
BMI	Branch N=0									30								-	-	-	-	-	-
BNE	Branch Z=0									D0								-	-	-	-	-	-
BPL	Branch N=0									10								-	-	-	-	-	-
BRA	Branch always									80								-	-	-	-	-	-
BVC	Branch V=0									50								-	-	-	-	-	-
BVS	Branch V=1									70								-	-	-	-	-	-
CLC	0→C								18									-	-	-	-	-	0
CLD	0→D								D8									-	-	0	-	-	-
CLI	0→I								58									-	-	-	0	-	-
CLV	0→V								B8									-	0	-	-	-	-
CMP	A-M	CD		0D	D9			C9				C5	C1	D5		D2	D1	N	-	-	-	Z	C
CPX	X-M	EC						E0				E4						N	-	-	-	Z	C
CPY	Y-M	CC						C0				C4						N	-	-	-	Z	C
DEC	Decrement	CE		DE			3A					C6		D6				N	-	-	-	Z	-
DEX	X-1→X								CA									N	-	-	-	Z	-
DEY	Y-1→Y								88									N	-	-	-	Z	-
EOR	A⊕M→A	4D		5D	59			49				45	41	55		52	51	N	-	-	-	Z	-
INC	Increment	EE		FE			1A					E6		F6				N	-	-	-	Z	-
INX	X+1→X								E8									N	-	-	-	Z	-
INY	Y+1→Y								C8									N	-	-	-	Z	-
JMP	Jump to new location	4C	7C			6C												-	-	-	-	-	-

Instructions	Operation # immediate data ~ NOT ^ AND v OR ⊕ Exclusive OR	Addressing modes																Status register					
		a	(a,x)	a,x	a,y	(a)	A	#	i	r	s	zp	(zp,x)	zp,x	zp,y	(zp)	(zp),y	N	V	D	I	Z	C
JSR	Jump to subroutine	20																N	-	-	-	Z	-
LDA	M→A	AD		BD	B9			A9				A5	A1	B5		B2	B1	N	-	-	-	Z	-
LDX	M→X	AE			BE			A2				A6			B6			N	-	-	-	Z	-
LDY	M→Y	AC			BC			A0				A4		B4				N	-	-	-	Z	-
LSR	0→MSB~LSB→C	4E			5E		4A					46		56				0	-	-	-	Z	C
NOP	No operation								EA									-	-	-	-		-
ORA	AvM→A	0D		1D	19			09				05	01	15		12	11	N	-	-	-	Z	-
PHA	A→Ms, S-1→S										48							-	-	-	-	-	-
PHP	P→Ms, S-1→S										08							-	-	-	-	-	-
PHX	X→Ms, S-1→S										DA							-	-	-	-	-	-
PHY	Y→Ms, S-1→S										5A							-	-	-	-	-	-
PLA	S+1→S, Ms→A										68							N	-	-	-	Z	-
PLP	S+1→S, Ms→P										28							N	V	D	I	Z	C
PLX	S+1→S, Ms→X										FA							N	-	-	-	Z	-
PLY	S+1→S, Ms→Y										7A							N	-	-	-	Z	-
ROL	C←MSB~LSB←C	2E		3E			2A					26		36				N	-	-	-	Z	C
ROR	C→MSB~LSB→C	6E		7E			6A					66		76				N	-	-	-	Z	C
RTI	Return from interrupt										40							N	V	D	I	Z	C
RTS	Return from subroutine										60							-	-	-	-		-
SBC	A-M-(~C)→A	ED		FD	F9			E9				E5	E1	F5		F2	F1	N	V	-	-	Z	C
SEC	1→C								38									-	-	-	-	-	1
SED	1→D								F8									-	-	1	-	-	-
SEI	1→I								78									-	-	-	1	-	-
STA	A→M	8D		9D	99							85	81	95		92	91	-	-	-	-	-	-
STX	X→M	8E										86			96			-	-	-	-	-	-
STY	Y→M	8C										84		94				-	-	-	-	-	-
STZ	00→M	9C		9E								68		74				-	-	-	-	-	-
TAX	A→X								AA									N	-	-	-	Z	-
TAY	A→Y			BC					AB									N	-	-	-	Z	-
TRB	~A^M→M											14						-	-	-	-		-
TSB	AvM→A											04						-	-	-	-	Z	-
TSX	S→X								BA									N	-	-	-	Z	-
TXA	X→A								8A									N	-	-	-	Z	-
TXS	X→S								9A									-	-	-	-	-	-
TYA	Y→A								98									N	-	-	-	Z	-

9 SFR (Special Function Register) Description

9.1 Index Table of SFR

Address	Symbol	Description
0x0000	**INTEN**	Interrupt Enable Register
0x0001	**SMCR**	System Mode Control Register
0x0002	**T0CTRL**	Timer 0 Control Register
0x0003	**T1CTRL**	Timer 1 Control Register
0x0004	**TMR0L**	Timer 0 Buffer Low Byte
0x0005	**TMR0H**	Timer 0 Buffer High Byte
0x0006	**TMR1L**	Timer 1 Buffer Low Byte
0x0007	**TMR1H**	Timer 1 Buffer High Byte
0x0008	**SECOND**	The SECOND counting of RTC
0x0009	**ALMSEC**	The Alarm specified of Second
0x000A	**MINUTE**	The MINUTE counting of RTC
0x000B	**ALMMIN**	The Alarm specified of Minute
0x000C	**HOUR**	The HOUR counting of RTC
0x000D	**ALMHR**	The Alarm specified of Hour
0x000E	**ADCCTRL**	ADC control Register
0x000F	**ADCDATA**	ADC DATA Register
0x0010	**Reserved**	**Reserved**
0x0011	**SMCR2**	System Mode Control Register2
0x0012	**UARTCKL**	UART Clock Low Byte
0x0013	**UARTCKH**	UART Clock High Byte
0x0014	**UARTCTRL**	UART Control register
0x0015	**UARTDATA**	UART Data input/output port
0x0016	**SPICTRL**	SPI Flash Control Register
0x0017	**SPIDATA**	SPI Flash Data Register
0x0018	**PAF**	Port A Flag
0x0019	**PACN**	Port A CMOS/NMOS control
0x001A	**PADIR**	Port A Direction Control
0x001B	**PADATA**	Port A Data Port
0x001C	**PBF**	Port B Flag
0x001D	**PBCN**	Port B CMOS/NMOS Control
0x001E	**PBDIR**	Port B Direction Control
0x001F	**PBDATA**	Port B Data Port

9.2 SFR Table

ADDRESS	NAME	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initialize value
0x0000	INTEN	UARTINT	URSTINT	ADCINT	T1INT	T0INT	PBINT	PAINT	RTCINT	*00000000*
0x0001	SMCR	--	EN32	SLEEP	LATCH	--	CPOFF	SYSCK1	SYSCK0	*x100x100*
0x0002	T0CTRL	T0F	T0AUTO	T0EN	T0CKS	BUZEN	--	--	--	*00000xxx*
0x0003	T1CTRL	T1F	T1AUTO	T1EN	T1CKS	--	--	--	--	*0000xxxx*
0x0004	TMR0L	Setting the counting buffer of Timer0								*xxxxxxxx*
x0x0005	TMR0H									*xxxxxxxx*
0x0006	TMR1L	Setting the counting buffer of Timer1								*xxxxxxxx*
0x0007	TMR1H									*xxxxxxxx*
0x0008	SECOND	--	--	Second of RTC (Hexadecimal)						*00xxxxxx*
0x0009	ALMSEC	2HzF	2HzEN	Second of Alarm Time (Hexadecimal)						*x0xxxxxx*
0x000A	MINUTE	--	--	Minute of RTC (Hexadecimal)						*00xxxxxx*
0x000B	ALMMIN	DAYF	DAYEN	Minute of Alarm Time (Hexadecimal)						*x0xxxxxx*
0x000C	HOUR	--	--	--	Hour of RTC (Hexadecimal)					*000xxxxx*
0x000D	ALMHR	ALMF	ALMEN	--	Hour of Alarm Time (Hexadecimal)					*x0xxxxxx*
0x000E	ADCCTRL	ADCen	ADCGO	TMPVDD	--	--	--	--	--	*000xxxxx*
0x000F	ADCDATA	ADCD [7]	ADCD [6]	ADCD [5]	ADCD [4]	ADCD [3]	ADCD [2]	ADCD [1]	ADCD [0]	*xxxxxxxx*
0x0010	Reserved									
0x0011	SMCR2	--	--	LVR	--	--	V1D5ADJ	CPVPPen	CPX5en	*xx1xx000*
0x0012	UARTCKL	Setting the counting buffer of ISO-7816 Clock								*xxxxxxxx*
0x0013	UARTCKH									*xxxxxxxx*
0x0014	UARTCTRL	--	SDPH	CLKCN	RSTCN	RSTD	UARTen	RT	MODE	*x0000000*
0x0015	UARTDATA	ISO-7816 Data Input/Output Port								*xxxxxxxx*
0x0016	SPICTRL	READ	SPITYPE	SPIen	--	--	--	SCK [1]	SCK [0]	*x00xxxxx*
0x0017	SPIDATA	SPI Data Input/Output Port								*xxxxxxxx*
0x0018	PAF/PATE	PAF [7]	PAF [6]	PAF [5]	PAF [4]	PAF [3]	PAF [2]	PAF [1]	PAF [0]	*00000000*
0x0019	PACN/PAIE	PACN [7]	PACN [6]	PACN [5]	PACN [4]	PACN[3]	PACN [2]	PACN [1]	PACN [0]	*00000000*
0x001A	PADIR	PADIR [7]	PADIR [6]	PADIR [5]	PADIR [4]	PADIR[3]	PADIR [2]	PADIR [1]	PADIR [0]	*00000000*
0x001B	PADATA/PH	PA [7]	PA [6]	PA [5]	PA [4]	PA [3]	PA [2]	PA [1]	PA [0]	*00000000*
0x001C	PBF/PBTE	PBF [7]	PBF [6]	PBF [5]	PBF [4]	PBF [3]	PBF [2]	PBF [1]	PBF [0]	*00000000*
0x001D	PBCN/PBIE	PBCN [7]	PBCN [6]	PBCN [5]	PBCN [4]	PBCN[3]	PBCN [2]	PBCN [1]	PBCN [0]	*00000000*
0x001E	PBDIR	PBDIR [7]	PBDIR [6]	PBDIR [5]	PBDIR [4]	PBDIR[3]	PBDIR [2]	PBDIR [1]	PBDIR [0]	*00000000*
0x001F	PBDATA/PH	PB [7]	PB [6]	PB [5]	PB [4]	PB [3]	PB [2]	PB [1]	PB [0]	*00000000*

10 Detail of Controlling SFR

10.1 Interrupt Control Register & Vector

0x000: INTEN (Interrupt Enable Register)

Address	0x0000	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Read or Write		UARTINT	URSTINT	ADCINT	T1INT	T0INT	PBINT	PAINT	RTCINT
Default Value (Reset)		0	0	0	0	0	0	0	0

This register controls 8 interrupt sources. The circuit will determine the priority in one of seven interrupt sources while two or more interrupt active. Bit7 is the highest priority, and Bit0 is the lowest priority. When interrupt happened that system will select correspondence interrupt vector to be executed automatically, and the interrupt vectors are shown in the following table. The vector of system reset is located in $0x7FFC and $0x7FFD.

Priority	Address	Interrupt source
0 (Highest)	0x7FE0 ~ 0x7FE1	UARTINT
1	0x7FE2 ~ 0x7FE3	URSTINT
2	0x7FE4 ~ 0x7FE5	ADCINT
3	0x7FE6 ~ 0x7FE7	T1INT
4	0x7FE8 ~ 0x7FE9	T0INT
5	0x7FEA ~ 0x7FEB	PBINT
6	0x7FEC ~ 0x7FED	PAINT
7 (Lowest)	0x7FEE ~ 0x7FEF	RTCINT

10.1.1 RTC Interrupt

0x000: INTEN (Interrupt Enable Register)

Address	0x0000	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Read or Write		UARTINT	URSTINT	ADCINT	T1INT	T0INT	PBINT	PAINT	RTCINT
Default Value (Reset)		0	0	0	0	0	0	0	0

INTEN.0 RTCINT (RTC interrupt switch, logic high active)
Write this bit
RTCINT=1, Allow RTC interrupt to occur
RTCINT=0, Disable RTC interrupt

RTC function includes Hour / Minute / Second units and also 2Hz / Day / Timer alarm flags. The setting RTC interrupt here is independence with RTC function. If user requests for RTC interrupt that follows the steps below. CPU will fetch 0x7FEE & 0x7FEF to program counter for the next instruction.

1. To make sure 32Khz clock had already been acting.
2. Waiting for the stability about 10us after 32Khz clock acting
3. Chose one even all of these functions to active (2HzEN, DayEN, ALMEN,)
4. Finally, to set RTCINT = 1 and then RTC interrupt will be executed automatically.

2Hz interrupt：

0x0009: ALMSEC (Alarm Second Register)

Address	0x0009	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Read or Write		2HzF	2HzEN	ASEC[5]	ASEC[4]	ASEC[3]	ASEC[2]	ASEC[1]	ASEC[0]
Default Value (Reset)		---	0	---	---	---	---	---	---

ALMSEC.6 2HzEN (2Hz trigger switch, logic high active)
Write this bit
2HzEN = 1 , enable 2Hz trigger.
2HzEN = 0 , disable 2Hz trigger.

After, four steps in 10.1.1 had to be done. If 2HzEN=1 and then 2Hz interrupt will occur periodically till 2HzEN=0 or RTCINT=0

ALMSEC.5 ~ 0 ASEC.5 ~ 0 (Setting the alarm value of second unit)
Write these bits
The range of alarm can set up from 00 ~ 59 which represent as 000000 ~ 111011 in binary unit.

Daily interrupt：

0x000B: ALMMIN (Alarm Minute Register)

Address	0x000B	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Read or Write		DAYF	DAYEN	AMIN[5]	AMIN[4]	AMIN[3]	AMIN[2]	AMIN[1]	AMIN[0]
Default Value (Reset)		---	0	---	---	---	---	---	---

ALMMIN.6 DAYEN (Daily trigger switch, logic high active)
Write this bit
DAYEN=1 , enable daily trigger. Daily trigger will happen when the RTC goes by from 23:59:59 to 00:00:00.
DAYEN=0 , disable daily trigger.

After, four steps in 10.1.1 had to be done. If DAYEN=1 and then Daily interrupt will occur periodically till DAYEN=0 or RTCINT=0

ALMMIN.5 ~ 0 AMIN.5 ~ 0 (Setting the alarm value of minute unit)
Write these bits
The range of alarm can set up from 00 ~ 59 which represent as 000000 ~ 111011 in binary unit.

Time alarmed interrupt：

0x000D: ALMHR (Alarm Hour Register)

Address	0x000D	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Read or Write		ALMF	ALMEN	---	AHR[4]	AHR[3]	AHR[2]	AHR[1]	AHR[0]
Default Value (Reset)		---	0	---	---	---	---	---	---

ALMHR.6 ALMEN (Time alarmed interrupt enabled, active high)
Write this bit
ALMEN=1 , enable time alarm trigger.
ALMEN=0 , disable time alarm trigger.

First, define the alarmed timing in these three registers
$0x0009h.bit5~0 , to define second in binary form
$0x000Bh.bit5~0, to define minute in binary form
$0x000Dh.bit4~0, to define hour in binary form

Make sure these four steps in 10.1.1 had to be done. Finally, the time alarmed interrupt will occur at the definition timing

ALMHR.4 ~ 0 AHR.4 ~ 0 (Setting the alarmed value of hour unit)
Write these bits
The range of alarm can set up from 00 ~ 23 which represent as 00000 ~ 10111 in binary unit.

10.1.2 Port A Interrupt

0x000: INTEN (Interrupt Enable Register)

Address	0x0000	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Read or Write		UARTINT	URSTINT	ADCINT	T1INT	T0INT	PBINT	PAINT	RTCINT
Default Value (Reset)		0	0	0	0	0	0	0	0

INTEN.1 PAINT (Port A interrupt switch, logic high active)
Write this bit
PAINT=1, Allow Port A interrupt to occur
PAINT=0, Disable Port A interrupt

Either rising or falling edge could trigger the Port A interrupt after PAINT had been setting to “1”. After Port A interrupt happened that CPU will fetch 0x7FEC & 0x7FED to program counter for the next instruction.

10.1.3 Port B interrupt

0x000: INTEN (Interrupt Enable Register)

Address	0x0000	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Read or Write		UARTINT	URSTINT	ADCINT	T1INT	T0INT	PBINT	PAINT	RTCINT
Default Value (Reset)		0	0	0	0	0	0	0	0

INTEN.2 PBINT (Port B interrupt switch, logic high active)
Write this bit
PBINT=1, Allow Port B interrupt to occur
PBINT=0, Disable Port B interrupt

Either rising or falling edge could trigger the Port B interrupt after PBINT had been setting to “1”. After Port B interrupt happened that CPU will fetch 0x7FEA & 0x7FEB to program counter for the next instruction.

10.1.4 Timer0 interrupt

0x000: INTEN (Interrupt Enable Register)

Address	0x0000	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Read or Write		UARTINT	URSTINT	ADCINT	T1INT	T0INT	PBINT	PAINT	RTCINT
Default Value (Reset)		0	0	0	0	0	0	0	0

INTEN.3 T0INT (Timer0 interrupt switch, logic high active)
Write this bit
T0INT=1, Allow timer0 interrupt to occur
T0INT=0, Disable timer0 interrupt

Either 32khz or 3.5Mhz could be the clock of Timer0. The setting here is independence with timer0 function. The Timer0 interrupt will happen after these setting up T0INT=1 & $0x0002h.bit5=1. Then, CPU will fetch 0x7FE8 & 0x7FE9 to program counter for the next instruction.

Defining timer0 counting buffer：
0x0004: TMR0L (Timer0 Low Byte Register)

Address	0x0004	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Write		TMR0L[7]	TMR0L[6]	TMR0L[5]	TMR0L[4]	TMR0L[3]	TMR0L[2]	TMR0L[1]	TMR0L[0]
Default value (Reset)		---	---	---	---	---	---	---	---
Read		TMR0L[7]	TMR0L[6]	TMR0L[5]	TMR0L[4]	TMR0L[3]	TMR0L[2]	TMR0L[1]	TMR0L[0]

TMR0L.7~0 TMR0L is the low byte counting buffer.
Define TMR0L= FFh

0x0005: TMR0H (Timer0 High Byte Register)

Address	0x0005	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Write		TMR0H[7]	TMR0H[6]	TMR0H[5]	TMR0H[4]	TMR0H[3]	TMR0H[2]	TMR0H[1]	TMR0H[0]
Default value (Reset)		---	---	---	---	---	---	---	---
Read		TMR0H[7]	TMR0H[6]	TMR0H[5]	TMR0H[4]	TMR0H[3]	TMR0H[2]	TMR0H [1]	TMR0H[0]

TMR0H.7~0 TMR0H is the high byte counting buffer.
Define TMR0H= 00h

We define timer0 counting buffer = 00FFh. High byte 00 represent TMR0H and low byte FF is TMR0L.
How to calculate the frequency of timer0 interrupt that we describe in following content.
Timer0 clock / (high byte * low byte) = interrupt frequency. So, we got the interrupt frequ3.5Mhz / (00FF=256) = 14Khz.

Setting timer0 control register：

0x0002: T0CTRL (Timer0 Control Register)

Address	0x0002	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Read or Write		T0F	T0AUTO	T0EN	T0CKS	BUZEN	---	---	---
Default Value (Reset)		0	0	0	0	0	---	---	---

T0CTRL.6 T0AUTO (Timer0 auto reload mode, high active)
Write this bit
T0AUTO=1 , timer0 will enter auto reload mode.
T0AUTO=0 , timer0 auto reload mode will disable.

T0CTRL.5 T0EN (Timer0 enabled, high active)
Write this bit
T0EN=1 , enable timer0 counting from register TMR0L and TMR0H.
T0EN=0 , disable timer0 counting.

T0CTRL.4 T0CKS (Timer0 clock source)
Write this bit
T0CKS=1 , Select external 32Khz crystal to be clock.
T0CKS=0 , Select internal 3.5Mhz to be clock.

10.1.5 Timer1 interrupt

0x000: INTEN (Interrupt Enable Register)

Address	0x0000	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Read or Write		UARTINT	URSTINT	ADCINT	T1INT	T0INT	PBINT	PAINT	RTCINT
Default Value (Reset)		0	0	0	0	0	0	0	0

INTEN.4 T1INT (Timer1 interrupt switch, logic high active)
Write this bit
T1INT=1, Allow timer1 interrupt to occur
T1INT=0, Disable timer1 interrupt

Either 32khz or 3.5Mhz could be the clock of Timer0. The setting here is independence with timer1 function. The Timer1 interrupt will happen after these setting up T1INT=1 & $0x0003h.bit5=1. Then, CPU will fetch 0x7FE6 & 0x7FE7 to program counter for the next instruction.

Defining timer1 counting buffer：

0x0006: TMR1L (Timer1 Low Byte Register)

Address	0x0006	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Write		TMR1L[7]	TMR1L[6]	TMR1L[5]	TMR1L[4]	TMR1L[3]	TMR1L[2]	TMR1L[1]	TMR1L[0]
Default value (Reset)		---	---	---	---	---	---	---	---
Read		TMR1L[7]	TMR1L[6]	TMR1L[5]	TMR1L[4]	TMR1L[3]	TMR1L[2]	TMR1L[1]	TMR1L[0]

TMR1L.7~0 TMR1L is the low byte counting buffer.
Define TMR1L= 00h

0x0007: TMR1H (Timer1 High Byte Register)

Address	0x0007	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Write		TMR1H[7]	TMR1H[6]	TMR1H[5]	TMR1H4]	TMR1H[3]	TMR1H[2]	TMR1H[1]	TMR1H[0]
Default value (Reset)		---	---	---	---	---	---	---	---
Read		TMR1H[7]	TMR1H[6]	TMR1H[5]	TMR1H[4]	TMR1H[3]	TMR1H[2]	TMR1H[1]	TMR1H[0]

TMR1H.7~0 TMR1H is the high byte counting buffer.
Define TMR1H= 1Fh

We define timer1 counting buffer = 00FFh. High byte 00 represent TMR1H and low byte FF is TMR1L.
How to calculate the frequency of timer0 interrupt that we describe in following content.
Timer1 clock / (high byte * low byte) = interrupt frequency. So, we got the interrupt frequency 3.5Mhz / (1F00=7936) = 462hz.

Setting timer1 control register：

0x0003: T1CTRL (Timer1 Control Register)

Address	0x0003	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Read or Write		T1F	T1AUTO	T1EN	T1CKS	---	---	---	---
Default Value (Reset)		0	0	0	0	---	---	---	---

T1CTRL.6 T0AUTO (Timer1 auto reload mode, high active)
Write this bit
T1AUTO=1 , timer1 will enter auto reload mode.
T1AUTO=0 , timer1 reload mode will disable.

T1CTRL.5 T1EN (Timer1 enabled, high active)
Write this bit
T1EN=1 , enable timer1 counting from register TMR1L and TMR1H.
T1EN=0 , disable timer1 counting.

T1CTRL.4 T1CKS (Timer1 clock source)
Write this bit
T1CKS=1 , Select external 32Khz crystal to be clock.
T1CKS=0 , Select internal 3.5Mhz to be clock.

10.1.6 ADC interrupt

0x000: INTEN (Interrupt Enable Register)

Address	0x0000	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Read or Write		UARTINT	URSTINT	ADCINT	T1INT	T0INT	PBINT	PAINT	RTCINT
Default Value (Reset)		0	0	0	0	0	0	0	0

INTEN.5 ADCINT (ADC interrupt switch, logic high active)
Write this bit
ADCINT=1, Allow A/D interrupt to occur
ADCINT=0, Disable A/D interrupt
The setting here allow ADC interrupt that A/D convert

0x000E: ADCCTRL (ADC Control Register)

Address	0x000E	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Read or Write		ADCen	ADCGO	TMPVDD	---	---	---	---	---
Default Value (Reset)		0	0	0	---	---	---	---	---

ADCCTRL.7 A/D convert enable / disable
Write
ADCen=1 A/D converter enable.
ADCen=0 A/D converter disable.

ADCCTRL.6 Trigger force A/D converting
Write low to high then A/D start to convert.
Read
ADCGO=1 , A/D in converting.
ADCGO=0 , A/D convert is finished.

ADCCTRL.5 Select A/D convert working voltage or temperature
Write
TMPVDD=0 ADC convert working voltage VDD.
TMPVDD=1 ADC convert temperature.

User have to notice that A/D function need follow the steps
(1) Control register bit7 ”ADCen”=High.
(2) Wait 100ms for A/D function ready.
(3) Setting Bit6 “ADCgo”=High , IC will start to A/D converting (2010/01/07 edit).

When ADC convert finished, the ADC would trigger interrupt
Register INTEN ($00H) bit5 must be enable (write high)
ADC interrupt vector is located at $7FE4H & $7FE5H

0x0001: SMCR (System Mode Control Register)

Address	0x0001	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Read or Write		---	-EN32	SLEEP	LATCH	---	CPOFF	SYSCK1	SYSCK0
Default Value (Reset)		---	1	0	0	---	1	0	0

SMCR.7 RESERVE

SMCR.6 -EN32 (Enable 32768 Hz crystal oscillation, low active)
32768 Hz crystal oscillator would be stop in some applications for power saving. Initialize default value would enable oscillation for saving the time of oscillate stable.

SMCR.5 SLEEP (Sleep mode, high active)
In this mode, main frequency oscillation would be stop, if all timing events are disabled. Those timing events are: timer 0 and 1。

CPU can be waked up by：PA, PB, UART (interrupt when ISO-7816 finish last bit transmission), URST (ISO-7816 will send a reset signal from master device to slave device) interrupt.

SMCR.4 LATCH (Latch all EPD outputs, active high pulse)
All EPD outputs would be updated when this bit is low. If user want to synchronize the output of Hi-V channel, it is necessary to set this bit high then low after Hi-V channel control register ($2000h ~ $203Ch) setting finished.

SMCR.3 RESERVE

SMCR.2 CPOFF (Charge pump OFF)
CPOFF=1 , disable charge pump.
CPOFF=0 , enable charge pump. Initialize default value is enabled.

SMCR.1~0 SYSCK [1:0] (System clock select)
The MF, main frequency, is around 4 MHz.

SYSCK1	0	0	1	1
SYSCK0	0	1	0	1
Clock	MF/8	MF/4	MF/2	MF

0x0002: T0CTRL (Timer0 Control Register)

Address	0x0002	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Read or Write		T0F	T0AUTO	T0EN	T0CKS	BUZEN	---	---	---
Default Value (Reset)		0	0	0	0	0	---	---	---

Timer0 is the 16-bit down count counter. It would stop counting when counter overflow or auto reload mode was not been presented. If auto reload mode enable, timer would reload the preset value and count down again.

T0CTRL.7 T0F (Timer0 overflow flag)
If timer0 occurred overflow, the T0F would latch high. Program should read this register to clear this bit before next counting. Program should preset TMR0L and TMR0H first and then set T0EN high to start counting. The preset data would be stop counting when T0EN is low.

T0CTRL.6 T0AUTO (Timer0 auto reload mode, high active)
Write
T0AUTO=1 , timer0 will enter auto reload mode.
T0AUTO=0 , timer0 auto reload mode will disable.

T0CTRL.5 T0EN (Timer0 enabled, high active)
Write
T0EN=1 , timer0 will start to count down from the value of TMR0L and TMR0H.
T0EN=0 , timer0 will stop counting.

T0CTRL.4 T0CKS (Timer0 clock source)
T0CKS=1 , Select external crystal frequency. (32768Hz).
T0CKS=0 , Select internal main frequency. (4MHz).

T0CTRL.3 BUZEN (Buzzer enabled, high active)
Write
BUZEN=1 , enable buzzer output. It's necessary to set Port B[1:0] in output mode.
BUZEN=1 , enable buzzer output. It's necessary to set Port B[1:0] in output mode.
BUZEN=0 , disable buzzer output. Port B [1:0] will return to I/O state.

T0CTRL.2~0 RESERVE

0x0003: T1CTRL (Timer1 Control Register)

Address	0x0003	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Read or Write		T1F	T1AUTO	T1EN	T1CKS	---	---	---	---
Default Value (Reset)		0	0	0	0	---	---	---	---

Timer1 is the 16-bit down count counter and build in isolation (80Khz) clock source. This clock source only for Timer1 to use. About the action of Timer1 would stop counting when counter overflow or reload mode was not been presented. If auto reload mode enable, timer would reload the preset value and count down again.

T1CTRL.7 T1F (Timer1 overflow flag)
If timer1 occurred overflow, the T1F would latch high. Program should read this register to clear this bit before start counting. Program should preset TMR1L and TMR1H first and then set T1EN high to start counting. The preset data would be stop counting when T1EN is low.

T1CTRL.6 T0AUTO (Timer1 auto reload mode, high active)
Write
T1AUTO=1 , timer1 will enter auto reload mode.
T1AUTO=0 , timer1 reload mode will disable.

T1CTRL.5 T1EN (Timer1 enabled, high active)
Write
T1EN=1 , timer1 will start to count down from the value of TMR1L and TMR1H.
T1EN=0 , timer1 will stop counting.

T1CTRL.4 T1CKS (Timer1 clock source)
T1CKS=1 , Select external crystal frequency. (32768Hz).
T1CKS=0 , Select internal main frequency. (4MHz).

T1CTRL.3~0 RESERVE

0x0004: TMR0L (Timer0 Low Byte Register)

Address	0x0004	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Write		TMR0L[7]	TMR0L[6]	TMR0L[5]	TMR0L[4]	TMR0L[3]	TMR0L[2]	TMR0L[1]	TMR0L[0]
Default value (Reset)		---	---	---	---	---	---	---	---
Read		TMR0L[7]	TMR0L[6]	TMR0L[5]	TMR0L[4]	TMR0L[3]	TMR0L[2]	TMR0L[1]	TMR0L[0]

TMR0L.7~0 Timer0 is the 16-bit counter. TMR0L is the low byte counting buffer.

0x0005: TMR0H (Timer0 High Byte Register)

Address	0x0005	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Write		TMR0H[7]	TMR0H[6]	TMR0H[5]	TMR0H[4]	TMR0H[3]	TMR0H[2]	TMR0H[1]	TMR0H[0]
Default value (Reset)		---	---	---	---	---	---	---	---
Read		TMR0H[7]	TMR0H[6]	TMR0H[5]	TMR0H[4]	TMR0H[3]	TMR0H[2]	TMR0H[1]	TMR0H[0]

TMR0H.7~0 TMR0H is the high byte counting buffer.

0x0006: TMR1L (Timer1 Low Byte Register)

Address	0x0006	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Write		TMR1L[7]	TMR1L[6]	TMR1L[5]	TMR1L[4]	TMR1L[3]	TMR1L[2]	TMR1L[1]	TMR1L[0]
Default value (Reset)		---	---	---	---	---	---	---	---
Read		TMR1L[7]	TMR1L[6]	TMR1L[5]	TMR1L[4]	TMR1L[3]	TMR1L[2]	TMR1L[1]	TMR1L[0]

TMR1L.7~0 Timer1 is the 16-bit counter. TMR1L is the low byte counting buffer.

0x0007: TMR1H (Timer1 High Byte Register)

Address	0x0007	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Write		TMR1H[7]	TMR1H[6]	TMR1H[5]	TMR1H[4]	TMR1H[3]	TMR1H[2]	TMR1H[1]	TMR1H[0]
Default value (Reset)		---	---	---	---	---	---	---	---
Read		TMR1H[7]	TMR1H[6]	TMR1H[5]	TMR1H[4]	TMR1H[3]	TMR1H[2]	TMR1H[1]	TMR1H[0]

TMR1H.7~0 TMR1H is the high byte counting buffer.
Before writing these registers, user has to disable timer0 / timer1 first.

0x0008: SECOND (SECOND Register)

Address	0x0008	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Read or Write		---	---	SEC[5]	SEC[4]	SEC[3]	SEC[2]	SEC[1]	SEC[0]

SECOND.7 ~ 6 RESERVE

SECOND.5 ~ 0 SEC.5 ~ 0 (Second register for RTC & perpetual calendar counting)

Writing for RTC second unit setting and the range from 00 ~ 59
In binary represent as 000000 ~ 111011.

Reading to obtain the current value of second unit.

0x000A: MINUTE (MINUTE Register)

Address	0x000A	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Read or Write		---	---	MIN[5]	MIN[4]	MIN[3]	MIN[2]	MIN[1]	MIN[0]

MINUTE.7 ~ 6 RESERVE

MINUTE.5 ~ 0 MIN.5 ~ 0 (Minute register for RTC & perpetual calendar counting)

Writing for RTC minute unit setting and the range from 00 ~ 59
In binary represent as 000000 ~ 111011.

Reading to obtain the current value of minute unit.

0x000C: HOUR (HOUR Register)

Address	0x000C	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Read or Write		---	---	---	HR[4]	HR[3]	HR[2]	HR[1]	NR[0]

HOUR.7 ~ 5 RESERVE

HOUR.4 ~ 0 HR.4 ~ 0 (Hour register for RTC & perpetual calendar counting)

Writing for RTC Hour unit setting and the range from 00 ~ 59
In binary represent as 000000 ~ 111011.

Reading to obtain the current value of hour unit.

The contents of RTC (Real Time Clock) register are SECOND, MINUTE, HOUR, time alarm and day-interrupt. Register SECOND, MINUTE and HOUR are designed for counting. They are calculating as hexadecimal number but store as binary number. Read the register as above to get the current value of RTC.

0x0009: ALMSEC (Alarm Second Register)

Address	0x0009	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Read or Write		2HzF	2HzEN	ASEC[5]	ASEC[4]	ASEC[3]	ASEC[2]	ASEC[1]	ASEC[0]

ALMSEC.7 2HzF (2Hz interrupt flag)
2HzF=1 , It means 2Hz interrupt has been triggered. It will be cleared after reading.
2HzF=0 , No 2Hz interrupt trigger.

ALMSEC.6 2HzEN (2Hz interrupt enabled, active high)
Write
2HzEN=1 , enable 2Hz interrupt. If 2HzEN is low, 2HzF would be low always.
2HzEN=0 , disable 2Hz interrupt.

ALMSEC.5 ~ 0 ASEC.5 ~ 0 (Setting the alarm value of second unit)
The range of alarm could be 00 ~ 59 in binary represent as 000000 ~ 111011.

0x000B: ALMMIN (Alarm Minute Register)

Address	0x000B	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Read or Write		DAYF	DAYEN	AMIN[5]	AMIN[4]	AMIN[3]	AMIN[2]	AMIN[1]	AMIN[0]

ALMMIN.7 DAYF (Day interrupt flag)
This bit would be set high, if RTC turn from 23:59:59 to 00:00:00. It will be cleared after reading.

ALMMIN.6 DAYEN (Day interrupt enabled, active high)
Write
DAYEN=1 , enable day interrupt. If DAYEN is low, DAYF would be low always. Day interrupt will happen when RTC turn from 23:59:59 to 00:00:00.
DAYEN=0 , disable day interrupt.

ALMMIN.5 ~ 0 AMIN.5 ~ 0 (Setting the alarm value of minute unit)
The range of alarm could be 00 ~ 59 in binary represent as 000000 ~ 111011.

0x000D: ALMHR (Alarm Hour Register)

Address	0x000D	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Read or Write		ALMF	ALMEN	---	AHR[4]	AHR[3]	AHR[2]	AHR[1]	AHR[0]

ALMHR.7 ALMF (Time alarm flag)
This bit would be set high, if RTC value is equal to alarm value. The alarm value could be set by ALMSEC, ALMMIN and ALMHR.

ALMHR.6 ALMEN (Time alarm interrupt enabled, active high)
Write
ALMEN=1 , enable alarm interrupt. If ALMEN is low, ALMF would be low always.
ALMEN=0 , disable alarm interrupt.

ALMHR.5 RESERVE

ALMHR.4 ~ 0 AHR.4 ~ 0 (Setting the alarm value of hour unit)
The range of alarm could be 00 ~ 23 in binary represent as 00000 ~ 10111.

0x000E: ADCCTRL (ADC Control Register)

Address	0x000E	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Read or Write		ADCen	ADCGO	TMPVDD	---	---	---	---	---
Default Value (Reset)		0	0	0	---	---	---	---	---

ADCCTRL.7 A/D convert enable / disable
Write
ADCen=1 A/D converter enable.
ADCen=0 A/D converter disable.

ADCCTRL.6 Trigger force A/D converting
Write low to high then A/D start to convert.
Read
ADCGO=1 , A/D in converting.
ADCGO=0 , A/D convert is finished.

ADCCTRL.5 Select A/D convert working voltage or temperature
Write
TMPVDD=0 ADC convert working voltage VDD.
TMPVDD=1 ADC convert temperature.

ADCCTRL.4 ~ 0 Reserve

User have to notice that A/D function need follow the steps
(4) Control register bit7 ”ADCen”=High.
(5) Wait 100ms for A/D function ready.
(6) Setting Bit6 “ADCgo”=High , IC will start to A/D converting (2010/01/07 edit).

When ADC convert finished, the ADC would trigger interrupt
Register INTEN ($00H) bit5 must be enable (write high)
ADC interrupt vector is located at $7FE4H & $7FE5H

0x000F: ADCDATA (ADC Data Register)

Address	0x000F	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Read or Write		ADCD[7]	ADCD6]	ADCD[5]	ADCD[4]	ADCD[3]	ADCD[2]	ADCD[1]	ADCD[0]
Default Value (Reset)		0	0	0	0	0	0	0	0

0x0011: SMCR2 (System Mode Control Register 2)

Address	0x0011	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Read or Write		---	---	LVR	---	---	V1D5ADJ	CPVPPen	CPX5en
Default Value (Reset)		---	---	1	---	---	0	0	0

SMCR2.5 LVR
Write
LVR=1 , enable low voltage reset.
LVR=0 , disable low voltage reset.

SMCR2.2 V1D5ADJ (V1D5 Bit)
Write
V1D5ADJ=1 , program setting will increase the voltage of V1D5.
V1D5ADJ=0 , keep the original state of V1D5.

SMCR2.1 CPVPPen (Charge pump VPP enabled bit)
Write
CPVPPen=1 , enable VPP voltage for charge pump.
CPVPPen=0 , disable VPP voltage.

SMCR2.0 CPX5en (Charge pump X5 enabled bit)
Write
CPX5en=1 , enable X5 (Vx5=15V) voltage on the charge pump.
CPX5en=0 , disable X5 voltage.

Note: Before entrance power saving ”Sleep” mode that user need disable LVR function first.

0x0012: UARTCKL (UART ClocK Low byte register)

Address	0x0012	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Write Only		UCK[7]	UCK[6]	UCK[5]	UCK[4]	UCK[3]	UCK[2]	UCK[1]	UCK[0]
Default Value (Reset)		---	---	---	---	---	---	---	---

UARTCKL.7 ~ 0 The low byte register of UART clock counting buffer.

0x0013: UARTCKH (UART Clock High byte register)

Address	0x0013	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Write Only		UCK[15]	UCK[14]	UCK[13]	UCK[12]	UCK[11]	UCK[10]	UCK[9]	UCK[8]
Default Value (Reset)		---	---	---	---	---	---	---	---

UARTCKH.7 ~ 0 The high byte register of UART clock counting buffer.

Write the value to UARTCKL and UARTCKH would determine the baud rate of UART.
Master Mode：The clock source is come from main frequency.
Slave Mode：The clock source is come from external ISO-7816 device.

Example：
Take 3.5Mhz as UART clock to calculate. Baud rate 19200 equal to 19.2Khz , 3584Khz / 187 = 19.4Khz, so program UARTCKL = #BBh ; UARTCKH = #00h.

0x0014: UARTCTRL (UART Control Register)

Address	0x0014	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Write		---	SDPH	CLKCN	RSTCN	RSTD	UARTen	RT	MODE
Default Value (Reset)		---	0	0	0	0	0	0	0
Read		PARITY	SDS	---	---	RSTS	---	---	---

UARTCTRL.7 PARITY (parity check)
Read only
PARITY=1 , It means odd parity comparison error.
PARITY=0 , It means odd parity comparison pass.

UARTCTRL.6 SDPH (Pull high selection of SD pin)
Write
SDPH=1 , To set pull high.
SDPH=0 , No pull high.
Read to get the state of SD pin

UARTCTRL.5 CLKCN (CMOS/NMOS selection of CLK pin)
Write
CKLCN=1 , Select CMOS type for CLK pin at output mode.
CKLCN=0 , Select NMOS type for CLK pin at output mode.

UARTCTRL.4 RSTCN (CMOS/NMOS selection of RST pin)
Write
RSTCN=1 , Select CMOS type for CLK pin at output mode.
RSTCN=0 , Select NMOS type for CLK pin at output mode.

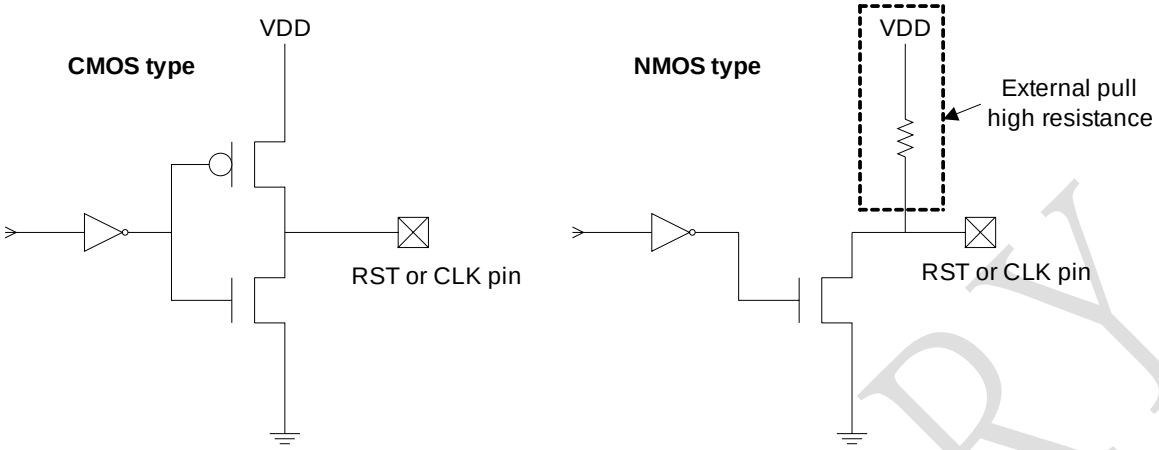
UARTCTRL.3 RSTD/RSTS (RST pin output/input)
Write
RSTD=1 , RST pin output high in master mode.
RSTD=0 , RST pin output low in slave mode
Read to get the RST pin state while in slave mode.

UARTCTRL.2 UARTen (UART enable, active high)
Write
UARTen=1 , enable UART interface circuit.
UARTen=0 , disable UART interface circuit.

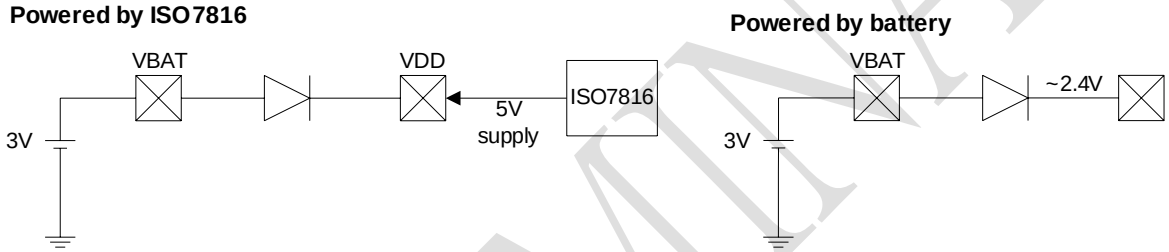
UARTCTRL.1 RT (Receive or transmit mode)
Write
RT=1 , enter data transmit mode.
RT=0 , enter data receive mode.

UARTCTRL.0 MODE (Master or slave mode)
Write
MODE=1 , enable master mode, clock source come from internal frequency.
MODE=0 , enable slave mode, clock source come from external ISO-7816 device.

Note: When using IS7816 function that user have to take care the state of (SD、RST、CLK pins):
SD: input pull high or floating, CMOS or NMOS output
RST: CMOS or NMOS output. (NMOS output need pull high resistance for logic high)
CLK: CMOS or NMOS output. (NMOS output need pull high resistance for logic high)



Note: Customer must notify if ISO7816 isn't needed that please connect (SD、RST、CLK pins) to GND in case of noise trigger IC from sleep mode.



Note: While connect to ISO7816 that IC powered by external 5V VDD. In order to avoid battery power fight with ISO7816 power PAD VBAT shouldn't short with VDD. Beside this application we suggest user short VBAT with VDD in external circuit.

0x0015: UARTDATA (UART DATA input/output port register)

Address	0x0015	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Write		TX[7]	TX[6]	TX[5]	TX[4]	TX[3]	TX[2]	TX[1]	TX[0]
Default Value (Reset)		---	---	---	---	---	---	---	---
Read		RX[7]	RX[6]	RX[5]	RX[4]	RX[3]	RX[2]	RX[1]	RX[0]

UARTDATA.7 ~ 0 TX / RX (This register access UART data transmit & receive)
Write a value to UARTDATA would force circuit to send out a byte to SD pin in transmitted mode.
In received mode, circuit would receive data from ISO-7816 device.

0x0016: SPICTRL (SPI interface Control Port)

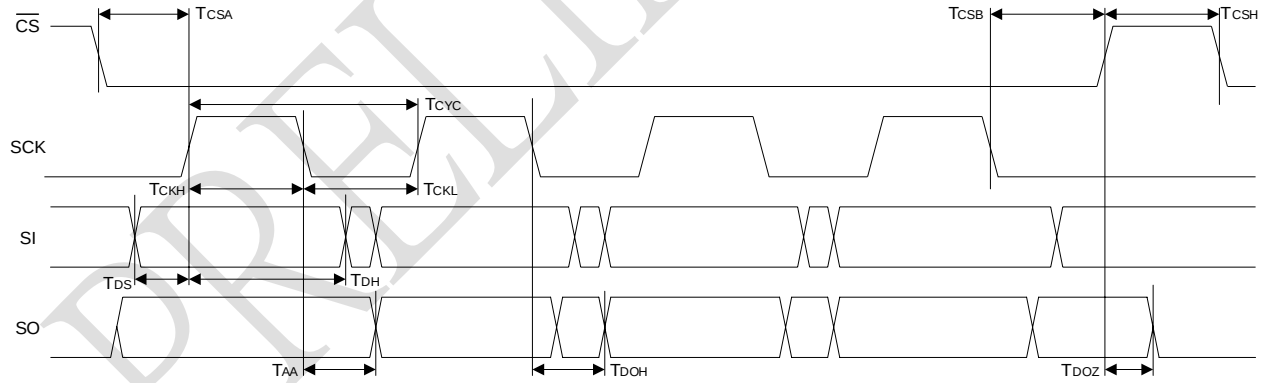
Address	0x0016	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Write		READ	SPItype	SPIen	---	---	---	SCK [1]	SCK [0]
Default Value (Reset)		---	---	---	---	---	---	---	---
Read		BUSY	SPItype	SPIen	---	---	---	---	---

SPICTRL.7 READ (W) / BUSY (R) (SPI flash read action enable bit and active status)
To write a high into this bit that would trigger the action of SPI flash reading. The chip would send out 8 clocks from PB[2], then receiving the data send by SPI flash. If chip is working on this procedure, then read this bit would be high. If chip has finished reading, then read this bit would get low.

SPICTRL.6 SPItype (specify the type of SPI waveform)
Write
SPItype=1 , SPI Specifies external device with following timing diagram.
This timing shows signal SPIDo is sent out by the negative trigger of SPIck, and should be captured by the positive trigger of SPIck.

SPItype=0 , SPI Specifies external device with following timing diagram.
This timing shows signal SPIDo is sent out by the positive trigger of SPIck, and should be captured by the negative trigger of SPIck.

SPI waveform



SPICTRL.5 SPIen (SPI interface circuit enable)
Write
SPIen=1 , enable SPI interface circuit.
SPIen=0 , disable SPI interface circuit.

SPICTRL.4 ~ 2 RESERVE

SPICTRL.1 SCK [1] (Clock Source Select Bit 1)

SPICTRL.0 SCK [1] (Clock Source Select Bit 0)
00: Φ0 / 2
01: Φ0 / 4
10: Φ0 / 8
11: Φ0 / 16

0x0017: SPIDATA (SPI interface DATA port)

Address	0x0017	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Write		SO [7]	SO [6]	SO [5]	SO [4]	SO [3]	SO [2]	SO [1]	SO [0]
Default Value (Reset)		---	---	---	---	---	---	---	---
Read		SI [7]	SI [6]	SI [5]	SI [4]	SI [3]	SI [2]	SI [1]	SI [0]

SPIDATA.7 ~ 0 Register for transmit / receive data via SPI interface

Register SPIDATA access SPI data for transmit or receive.
Write to this register would generate a byte of SPI clock out while SPI transmit mode.
Write high to SPICTRL.7 SPI clock also generate in receive mode. Then polling SPICTRL.7 if receive finished it will latch to low.

SPIDATAOUT share with Port B[3]
SPIDATAIN share with Port B[4]
SPICLOCK share with Port B[2]

0x0018: PAF (Port A Flag register)

Address	0x0018	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Write		PATE[7]	PATE[6]	PATE[5]	PATE[4]	PATE[3]	PATE[2]	PATE[1]	PATE[0]
Default Value (Reset)		0	0	0	0	0	0	0	0
Read		PAF[7]	PAF[6]	PAF[5]	PAF[4]	PAF[3]	PAF[2]	PAF[1]	PAF[0]

PAF.7 ~ 0 Register setting for system wake up.

The write process for PAF define as PATE[7:0]. Writing high to PATE[n] (n=7~0) that can wake up CPU by input a trigger signal (positive or negative trigger) on Port A[n] channel. PATE[n] would be activated while PADIR[n] is setting low as input mode.

The read process for PAF define as PAF[7:0]. Read PAF checking which channel of Port A pin has been triggered. A high logic of PAF[n] show that Port A[n] has been triggered. If PATE[n] is low, then PAF[n] would be low always and trigger Port A[n] would not change PAF[n].

0x0019: PACN (Port A CMOS/NMOS output driving control register)

Address	0x0019	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Write Only (PADIRn=0)		PAIE[7]	PAIE[6]	PAIE[5]	PAIE[4]	PAIE[3]	PAIE[2]	PAIE[1]	PAIE[0]
Write Only (PADIRn=1)		PACN[7]	PACN[6]	PACN[5]	PACN[4]	PACN[3]	PACN[2]	PACN[1]	PACN[0]
Default Value (Reset)		0	0	0	0	0	0	0	0

PACN.7 ~ 0 Register setting for Port A[n] output type & external trigger interrupt.

The write process for PACN has two functions:
(1) In this condition PADIR[n]=0, PACN represent as PAIE
PAIE[n]=1 means external trigger interrupt enable via Port A[n].
PAIE[n]=0 means external trigger interrupt disable.
(2) PADIR[n]=1 , PACN still the same name
PACN[n]=1 set Port A[n] as CMOS type
PACN[n]=0 set Port A[n] as NMOS type

0x001A: PADIR (Port A Direction control register)

Address	0x001A	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Write Only		PADIR[7]	PADIR[6]	PADIR[5]	PADIR[4]	PADIR[3]	PADIR[2]	PADIR[1]	PADIR[0]
Default Value (Reset)		0	0	0	0	0	0	0	0

PADIR.7 ~ 0 Register setting for Port A[n] input or output function.

PADIR[n]=1 set Port A[n] as output pin, PADIR[n]=0 set Port A[n] as input pin.
The initialize default value of PADIR[n] are low, it means PA[n] is input after reset. If user needs to set PA[n] as output pin, it is necessary to consider the short period of floating that would affect system stable before program changes the condition of Port A[n].

0x001B: PADATA (Port A DATA input/output port register)

Address	0x001B	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Write (PADIRn=0)		PAPH[7]	PAPH[6]	PAPH[5]	PAPH[4]	PAPH[3]	PAPH[2]	PAPH[1]	PAPH[0]
Write (PADIRn=1)		PA[7]	PA[6]	PA[5]	PA[4]	PA[3]	PA[2]	PA[1]	PA[0]
Default Value (Reset)		0	0	0	0	0	0	0	0
Read		PA[7]	PA[6]	PA[5]	PA[4]	PA[3]	PA[2]	PA[1]	PA[0]

PADATA.7 ~ 0 Register access I/O state of Port A[n].

Writing register PADATA has two functions:
(1) While PADIR[n]=0 , PADATA represent as PAPH
PAPH[n]=1 , enable pull high resistor. This function would be activated when PADIR[n] is low.
(2) While PADIR[n]=1 , PADATA represent as PA
Write
PA[n]=1 , Port A channel will output logic high, PA[n]=0 , Port A channel will output logic low.
Read
To obtain the state of Port A[n] channel , Logic high or low.

0x001C: PBF (Port B Flag register)

Address	0x001C	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Write		PBTE[7]	PBTE[6]	PBTE[5]	PBTE[4]	PBTE[3]	PBTE[2]	PBTE[1]	PBTE[0]
Default Value (Reset)		0	0	0	0	0	0	0	0
Read		PBF[7]	PBF[6]	PBF[5]	PBF[4]	PBF[3]	PBF[2]	PBF[1]	PBF[0]

PAF.7 ~ 0 Register setting for system wake up.

The write process for PBF define as PBTE[7:0]. Writing high to PBTE[n] (n=7~0) that can wake up CPU by input a trigger signal (positive or negative trigger) on Port B[n] channel. PBTE[n] would be activated while PBDIR[n] is setting low as input mode.

The read process for PBF define as PBF[7:0]. Read PBF checking which channel of Port B pin has been triggered. A high logic of PBF[n] show that Port B[n] has been triggered. If PBTE[n] is low, then PBF[n] would be low always and trigger Port B[n] would not change PBF[n].

0x001D: PBCN (Port B CMOS/NMOS output driving control register)

Address	0x001D	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Write Only (PBDIRn=0)		PBIE[7]	PBIE[6]	PBIE[5]	PBIE[4]	PBIE[3]	PBIE[2]	PBIE[1]	PBIE[0]
Write Only (PBDIRn=1)		PBCN[7]	PBCN[6]	PBCN[5]	PBCN[4]	PBCN[3]	PBCN[2]	PBCN[1]	PBCN[0]
Default Value (Reset)		0	0	0	0	0	0	0	0

PBCN.7 ~ 0 Register setting for Port B[n] output type & external trigger interrupt.

The write process for PBCN has two functions:
(1) In this condition PBDIR[n]=0 , PBCN represent as PBIE
PBIE[n]=1 means external trigger interrupt enable via Port B[n] .
PBIE[n]=0 means external trigger interrupt disable.
(2) PBDIR[n]=1 , PBCN still the same name
PBCN[n]=1 set Port B[n] as CMOS type
PBCN[n]=0 set Port B[n] as NMOS type

0x001E: PBDIR (Port B Direction control register)

Address	0x001E	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Write Only		PBADIR[7]	PBDIR[6]	PBDIR[5]	PBDIR[4]	PBDIR[3]	PBDIR[2]	PBDIR[1]	PBDIR[0]
Default Value (Reset)		0	0	0	0	0	0	0	0

PBDIR.7 ~ 0 Register setting for Port B[n] input or output function.

PBDIR[n]=1 set Port B[n] as output pin, PBDIR[n]=0 set Port B[n] as input pin.
The initialize default value of PBDIR[n] are low, it means PB[n] is input after reset. If user needs to set PB[n] as output pin, it is necessary to consider the short period of floating that would affect system stable before program changes the condition of Port B[n].

0x001F: PBDATA (Port B Data input/output port register)

Address	0x001F	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Write (PBDIRn=0)		PBPH[7]	PBPH[6]	PBPH[5]	PBPH[4]	PBPH[3]	PBPH[2]	PBPH[1]	PBPH[0]
Write (PBDIRn=1)		PB[7]	PB[6]	PB[5]	PB[4]	PB[3]	PB[2]	PB[1]	PB[0]
Default Value (Reset)		0	0	0	0	0	0	0	0
Read		PB[7]	PB[6]	PB[5]	PB[4]	PB[3]	PB[2]	PB[1]	PB[0]

PBDATA.7 ~ 0 Register access I/O state of Port B[n].

Writing register PBDATA has two functions:
(1) While PBDIR[n]=0 , PBDATA represent as PBPH
PBPH[n]=1 , enable pull high resistor. This function would be activated when PBDIR[n] is low.
(2) While PBDIR[n]=1 , PBDATA represent as PB
Write
PB[n]=1 , Port B channel will output logic high, PB[n]=0 , Port B channel will output logic low.
Read
To obtain the state of Port B[n] channel , Logic high or low.

11 Absolute Maximum Ratings

Symbol	Parameter	Rating	Unit
VDD	Logic Supply voltage	-0.5 to +7	V
Vin	Logic Input voltage	-0.5 to VDD+0.5	V
Vout	Logic Output voltage	-0.5 to VDD+0.5	V
Topr	Operation temperature range	-20 to -+85	°C
Tstg	Storage temperature range	-65 to 150	°C

12 Operating Ratings

Symbol	Description	Value			Unit
		Min	Typ	Max	
VDD	Working voltage	2	3	5.5	V
V0i	Internal supply Hi voltage	36		42	V
Vripple			200		mV
V0e	External supply Hi voltage	13		40	V
V1	Half voltage from V0		V0*1/2		V

13 DC Electrical Characteristics

item	Description	Condition	Symbol	Applicable pin	Value			Unit
					Min	Typ	Max	
Input high logic voltage			V_{IH}			0.7*VDD		V
Input low logic voltage			V_{IL}			0.2*VDD		V
Output high logic voltage		I_{OH} = -0.4mA	V_{OH}			VDD-0.4		V
Output low logic voltage		I_{OL} = 0.4mA	V_{OL}			0.4		V
Input high leakage current		Vin=VDD	I_{LIH}			1		uA
Input low leakage current		Vin=GND	I_{LIL}			-1		uA
Hi-V channel1 impedance	segments output resistance			Y1 ~ Y120		1K	2K	ohm
Hi-V channel2 impedance	common /background output resistance			Y121 ~ Y122		50	100	ohm
Hi-V channels leakage current	Hi-V channel output off			Y1 ~ Y122		0.05	0.1	nA
Active Mode	MCU active , charge pump on	VDD=3V , CLK=4Mhz		VDD			400	uA
Power consumption Mode	Low speed clock , charge pump off	VDD=3V		VDD			50	uA
Sleep Mode	MCU & charge pump off	VDD=3V , No CLK		VDD			50	nA
Input terminal capacity		CLK=1Mhz , Ta=25°C					4	pF
I/O capacity			$C_{I/O}$				8	pF

14 Ordering Information

Part Number	Pin Count	Package
DM120C16EW	-	Wafer (Pb-Free)
DM120C16EWB	-	Wafer + Gold bump (Pb-Free)
DM120C16E	168	Dice (Pb-Free)
DM120C16EB	168	Dice + Gold bump (Pb-Free)
DM120C16EC	176	COF (Roll) (Pb-Free)
DM120C16EP	176	COF (Tray) (Pb-Free)

Disclaimer

The information appearing in this publication is believed to be accurate. Integrated circuits sold by DAVICOM Semiconductor are covered by the warranty and patent indemnification provisions stipulated in the terms of sale only. DAVICOM makes no warranty, express, statutory, implied or by description regarding the information in this publication or regarding the information in this publication or regarding the freedom of the described chip(s) from patent infringement. FURTHER, DAVICOM MAKES NO WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PURPOSE. DAVICOM reserves the right to halt production or alter the specifications and prices at any time without notice. Accordingly, the reader is cautioned to verify that the data sheets and other information in this publication are current before placing orders. Products described herein are intended for use in normal commercial applications. Applications involving unusual environmental or reliability requirements, e.g. military equipment or medical life support equipment, are specifically not recommended without additional processing by DAVICOM for such applications.

Please note that application circuits illustrated in this document are for reference purposes only.

DAVICOM's terms and conditions printed on the order acknowledgment govern all sales by DAVICOM. DAVICOM will not be bound by any terms inconsistent with these unless DAVICOM agrees otherwise in writing. Acceptance of the buyer's orders shall be based on these terms.

Company Overview

DAVICOM Semiconductor Inc. develops and manufactures integrated circuits for integration into data communication products. Our mission is to design and produce IC products that are the industry's best value for Data, Audio, Video, and Internet/Intranet applications. To achieve this goal, we have built an organization that is able to develop chipsets in response to the evolving technology requirements of our customers while still delivering products that meet their cost requirements.

Products

We offer only products that satisfy high performance requirements and which are compatible with major hardware and software standards. Our currently available and soon to be released products are based on our proprietary designs and deliver high quality, high performance chipsets that comply with modem communication standards and Ethernet networking standards. ***For customized products, please contact to Davicom's sales***

Contact Windows

For additional information about DAVICOM products, contact the Sales department at:

Headquarters

Hsin-chu Office:
No.6, Li-Hsin 6th Rd.,
Hsinchu Science Park,
Hsin-chu City 300, Taiwan, R.O.C.
TEL: +886-3-5798797
FAX: +886-3-5646929
MAIL: sales@davicom.com.tw
HTTP: http://www.davicom.com.tw

WARNING
Conditions beyond those listed for the absolute maximum may destroy or damage the products. In addition, conditions for sustained periods at near the limits of the operating ranges will stress and may temporarily (and permanently) affect and damage structure, performance and function.