

23 | 24

3c) i) master out, slave in

master in, slave out

purpose: to decide which device sets the common clock rate to be used.

master sets the clockrate.

SSI reg : SSIICR1 @ 0x4000.8004

set bit 2 to 1: slave mode } write 0x0000.0004 ✓  
bit 2 is 0: master mode } write 0x0000.0000 ✓

ii) possible SSI communication errors :

[TBC]

1. checksum error
2. timeout error
3. parity error

4. noise or distortion after clock edges

b) SSI3CLK: P00

SSI3 Rx: PD2

SSI3 Tx: PD3

J connector block: J3

c) RCCGSSI: 0x400F.E61C  
RCCG GPIO: 0x400F.E608  
GPIO PCTRL: 0x4000.752C  
GPIO AFSEL: 0x4000.7420  
SSICR0: 0x4000.B000  
SSICR1: 0x4000.B004

Addr	Data
0x0000.0008	
0x0000.0008	
0x0000.1111 or 0000 if don't need FSD	1101 if don't need FSD same
0x0000.000F or 0000 if FSD	P00-3 = 0b1111 or P00,2,3 → 0b1101 = 0x0
0x0000.038F	DSS = 0xF = 0b1111, 0b1000-1111 SPH = 1, SPL = X, FRF = 00' → 0x8F
0x0000.0002	BR = $\frac{SCLK}{CPSDNSR(HSCR)}$ SPLK = $\frac{16M}{8(HSCR)}$ HSCR = $\frac{16 \times 10^3}{8 \times 512} = 3.906$ SCR ≈ 3 = 0x03

4. a)  $16/(32-6) \text{ timer } 3A = T3CCPD \Rightarrow PB2_{\text{off}}$

	<u>Addr</u>	<u>Data</u>
b)i) RCGCTIMER	0x400F.E604	0x0000.0008 ← Ob1000
RCGCGPIO	0x400F.E608	0x0000.0002 (Port B)
GPIOPCTL	0x4000.552C	0x0000.0700 (PB2 → 1)
GPIOPSEL	0x4000.5420	0x0000.0004 (PB2 = Ob0100)
GPTIMCFG	0x4003.3000	0x0000.0004

ii)

	<u>Addr</u>	<u>Data</u>	
NVIC_EN1	0xE000.E104	0x0000.0008	Int. No. 35 ⇒ EN1 (32-6) bit no. 3 ie Ob000...1000 = 0x0000.0008
NVIC_PRI8	0xE000.E420	0x6000.0000	

int. no. 35 : PRI8 ,  $35 = 4(8) + 3 \Rightarrow INT0$   
Set to 3:  $\frac{Ob0110...}{6}$

### c) Combine H-bridge + PWM drive

- direction control why two phs, ZN1 and ZN2 in complementary polarities. H-bridge with logic gates and transistors can effectively change motor direction by reversing the polarity of voltage outputs
- speed control why PWM signal or EN ph allows for a smaller PWM signal voltage to be amplified using a separate high voltage power source to provide power for the motor. This is preferable to using PWM directly on the actual power source, which is possible but less efficient.
- operating control ? ?
- safety limit

22|23

(1814)

$$\begin{array}{r}
 1. \text{ a) } 56 \div 2 = 28 \quad r \ 0 \\
 | & 14 \quad r \ 0 \\
 | & 7 \quad r \ 0 \\
 3 \quad r \ 1 \\
 1 \quad r \ 1 \\
 0 \quad r \ 1
 \end{array}$$

$$0.56 = 0.b\ 111000$$

$$\begin{array}{l}
 \text{b) } 0.63 \times 2 = 1.26 \\
 .26 \times 2 = 0.52 \\
 .52 \times 2 = 1.04 \\
 0.04 \times 2 = 0.08 \\
 0.08 \times 2 = 0.16 \\
 0.16 \times 2 = 0.32
 \end{array}$$

$$\therefore 0.63_{10} = 0.101000_2$$

$$\text{c) } 7986 \div 16 = 499 \quad r \ 2 \quad 2$$

$$\begin{array}{r}
 499 \div 16 = 31 \quad r \ 3 \quad 3 \\
 31 \div 16 = 1 \quad r \ 15 \quad F \\
 1 \div 16 = 0 \quad r \ 1 \quad 1
 \end{array}
 \quad \therefore 7986_{10} = 1F32_{16}$$

$$\begin{array}{r}
 \text{d) } -34.78 \quad \text{integer: } 34 \div 2 = 17 \quad r \ 0 \\
 17 \div 2 = 8 \quad r \ 1 \\
 8 \div 2 = 4 \quad r \ 0 \\
 4 \mid 2 = 2 \quad r \ 0 \\
 2 \mid 2 = 1 \quad r \ 0 \\
 1 \mid 2 = 0 \quad r \ 1
 \end{array}
 \quad \left. \right\} 34_{10} = 100010_2$$

$$\text{decimal: } 0.78 \times 2 = 1.56$$

$$0.56 \times 2 = 1.12$$

$$0.12 \times 2 = 0.24$$

$$0.24 \times 2 = 0.48$$

$$0.48 \times 2 = 0.96$$

$$0.96 \times 2 = 1.92 \quad \text{together,}$$

$$0.92 \times 2 = 1.84$$

$$0.84 \times 2 = 1.68$$

$$0.68 \times 2 = 1.36$$

$$1.36 \times 2 = 0.72$$

$$0.72 \times 2 = 1.44$$

$$1.44 \times 2 = 0.88$$

$$0.88 \times 2 = 1.76$$

$$1.76 \times 2 = 1.52$$

$$1.52 \times 2 = 1.04$$

$$1.04 \times 2 = 0.08$$

$$0.08 \times 2 = 0.16$$

$$0.16 \times 2 = 0.32$$

$$0.32 \times 2 = 0.64$$

$$0.64 \times 2 = 1.28$$

$$1.28 \times 2 = 0.56$$

$$0.56 \times 2 = 1.12$$

$$1.12 \times 2 = 0.24$$

$$0.24 \times 2 = 0.48$$

$$\therefore 0.78_{10}$$

$$= 0.11001111010111000101000$$

together,

$$34.78$$

$$= 10\overline{0010} \dots$$

$$= 1.00010\dots \times 2^5$$

sign: 1

$$\text{exponent: } 5 + 127 = 132$$

#

$$= 100001000\checkmark$$

23-bit mantissa: 0001 0110 0111 1010 1110 000

repeat



2. a) Conv. to dec : R1: 85

R2: 136

R3: 9

$$R1 \times R3 + R2 = 90 \text{ } 1_{10} = 385_{16}$$

$$\therefore R0 = 0x0385 \quad \checkmark$$

b)  $B = 0d16 = 0b1000$

$$\text{LSR } 3 : 0b1000 \rightarrow 0b10$$

$$R1 = 0b10 = 0d2$$

$$A, \text{LSL } 3 : 0d64 = 0b1000000$$

$$\text{LSL } 3 : 0b1000000 \rightarrow 0b1000\ 0000\ 00 = 0d512$$

$$0d512 + 0d2 = 0d514 \quad \checkmark$$

c)  $0xAFFA = 0b1010\ 1010\ 1010\ 1010$

!

$$0xBDD0 = 0b1011\ 1011\ 1011\ 1011$$

$$\therefore \text{MASK} = 0b\ 0001\ 0001\ 0001\ 0001$$

$$= 0x1111 \quad \checkmark$$

d)

?

8-bit

3. a) i) The bit mask is achieved by writing data to the GPIO Register address with a

12-bit offset. Bits 2:9 of the offset are used as a mask, where 1 will allow

the data written to overwrite the corresponding bits in GPIODATA while a 0 will block  
the data write, so GPIO DATA will remain unchanged.

S2C OFFSET...

ii) addr: 0x4002.5000 # col 2 : write 0x0000 0022 #

b)

	<u>UART</u>	<u>SSZ</u>
Fxn & Features:	Tx, Rx, GND Configurable data format & transmission speed Asynchronous; supports parallel	MISO, MOSI, SS, CLK, GND Synchronous; Master Slave Serial only

Errors:	baud rate mismatch data format mismatch e.g. big-endian vs little-endian overrun error?	- parity error - ; clock-skew error
---------	--	--

Applications:	only 2 slaves e.g.? Bluetooth module GPS module	multiple slaves e.g. diff robotics servos
---------------	---	---

c)

	<u>Addr</u>	<u>Data</u>	$400 \text{ Kbps} = \frac{40 \text{ MHz}}{\text{CPSDVSR}(\text{Clock})}$
SSICRD	0x4000.A900	0x 0000.0986 ✓	$\downarrow$
SSICPSR	0x4000.A010	0x 0000.000A <del>X</del> ✓ ← carry	$\text{CPSDVSR}(\text{Clock}) = 100$ $\downarrow$ 10 $= 0xA$

7:4: 0b1000 = 0x8

3:0: 7-b7: 0b

4. a)

	<u>Addr</u>	<u>Data</u>	Timer 4A, 1b-8H
RCCGCTIMBR	0x 400F.E604	0x 0000.0010	$T4CCPDR \rightarrow 9C0$
RCCGCGPIO	0x 400F.E608	0x 0000.0104	AF #7, ph 0
GPIOPCTL	0x 4000.652C	0x 0000.0007	choose 1b-8H
GPTMCFC	0x 4003.4000	0x 0000.0004	(periodic)
GPTM7AMR	0x 4003.4004	0x 0000.0012	$\begin{array}{cccccccccc} 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ \hline 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \end{array}$ $= 0x12$

4.b)

	<u>Addr</u>	<u>Data</u>	
GPIO PCIL	(glwn)	0x0000.0055 //	M1PWM 0: PD0 } col 5
PWM1LOAD	0x4002.9050 //	0x0000.1387 //	1: PD1
PWM1CMPA	0x4002.9058 //	0x0000.05DB //	clock: 10MHz required: 2kHz
PWM1CMPB	0x4002.905C //	0x0000.0DAB //	need $\frac{10M}{2K} = 5000$ ticks per cycle $\therefore \text{load: } 4991 = 0x1387$

70% duty cycle:  $4999 - (5000 \times 70\%) = 1499 = 0x3DB$

30% duty cycle:  $4999 - (5000 \times 30\%) = 3499 = 0x8AB$

21/22

1. a)  $78 = 1001110$

$0.1234 \times 2 =$

(ww)

b)  $A=25 = 0b\ 11001$

$B=80 = 0b$

$A=80 \div 4 = 20$

$20 \times 2^9 = 10240 \neq$

c)  $0x0F$  bruh?

d)  $0xFF$  bruh?

e)  $0.3697 \times 16 = 5.9152$

$.9152 \times 16 = 14.6432$

$\times 16 = 10.2912$

$= 4.6592$

$= 10.5472$

.5EA4A ↴

2. a) R1: 1010 1011 1100 1101

R2: 0001 0000 0001 0000 = 0x0101 ↴

R1: 1011 1011 1101 1101

b)  $12 \times \frac{37}{4} = 56 = 500$

c) R1: 0xFFFF = ..... ↴

R2: Y

R2: ~Y

BCD = 1011 1100 0111 1101

~Y = BCD

Y = 0100 0011 1000 0010 = 0x4382 ↴

