Lab 8: LCD display and Interfacing ADC with 8051 using SPI protocol

1 Problem Statement

Write assembly language program to:

1. Add two 8 bit numbers:
   - Store any two 8-bit numbers in register R0 and memory location 40H, respectively.
   - Perform adddition of these two 8-bit numbers.
   - Store the result in memory location 42H and carry in memory location 44H.
   - Display the hex value of sum and carry on first and second line of the LCD.

Write C program to:

1. Use the sample program provided (`adc.c`) to measure variable voltage from potentiometer:
   (a) Connect potentiometer between 3.3V and ground supplies.
   (b) Use ADC to measure the voltage at variable point of potentiometer.
   (c) Display measured voltage on LCD in the format: "Voltage: —— mV".
2. Measure temperature using LM35 sensor:
   (a) Configure 8051 for serial communication with ADC.
   (b) Connect temperature sensor LM35 to ADC.
   (c) Use the setup to measure room temperature and display it on LCD

2 Serial Communication between Master and Slave with SPI

- A shift clock is generated by master and fed to slave device.
- When 8 clock pulses are provided, contents of master and slave shift registers are *exchanged*. Data transfer happens *simultaneously*.
- Data is shifted once when a write occurs in the master shift register.
- Here, output of the master shift register is connected to input of the slave through line 'Master Out Slave In (MOSI)' and output of the slave shift register is connected to the input of the master through line called 'Master In Slave Out (MISO)'.

3 ADC Interfacing with Pt-51

3.1 Connection Diagram

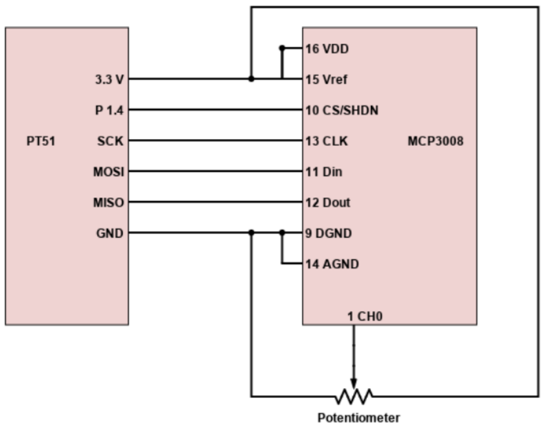

Figure 1: Pt51 interfacing with ADC and potential divider

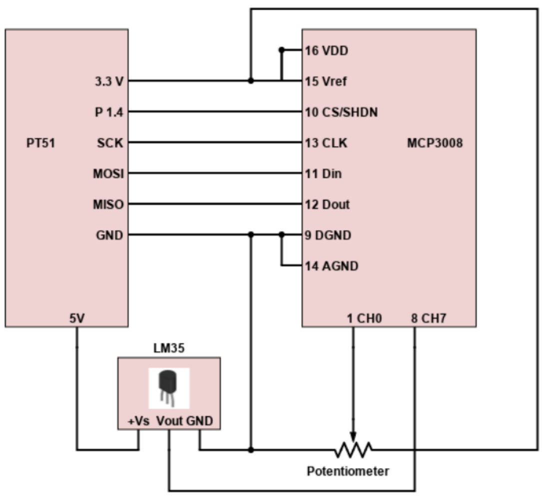

Figure 2: Pt51 interfacing with ADC and temperature sensor

3.2 Hardware Interface

- The ADC used for SPI interface is MCP 3008
- Analog input can be given between CH0 and AGND
- 10 bit digital output is available on D_{out} pin of ADC serially

- Microcontroller takes the input and modifies the data to show value on LCD display
- SPI pins on the Pt-51 that are used for connecting wires to the ADC are:

Pin	Function	Description
P1.4	SS_{bar}	Chip Select
P1.5	MISO	Serial Input
P1.6	SCK	Serial Clock
P1.7	MOSI	Serial Output

3.3 Steps for ADC interfacing with SPI protocol

1. Configure SPCON register as described below to enable SPI communication:
   (a) Free the SS pin for a general-purpose
   (b) Select one of the Master clock rates (By choosing appropriate values of SPR0, SPR1, SPR2 we can get different baud rates. Select the appropriate baud rate.)
   (c) Configure the SPI module as Master (In this case)
   (d) Selects serial clock polarity and phase (1,1) or (0,0)
   (e) Enable the SPI module (SPEN = 1)
   (f) Enable all the interrupts

7	6	5	4	3	2	1	0
SPR2	SPEN	SSDIS	MSTR	CPOL	CPHA	SPR1	SPR0

Bit Number	Bit Mnemonic	Description
7	SPR2	**Serial Peripheral Rate 2** Bit with SPR1 and SPR0 define the clock rate.
6	SPEN	**Serial Peripheral Enable** Cleared to disable the SPI interface. Set to enable the SPI interface.
5	SSDIS	**$\overline{SS}$ Disable** Cleared to enable $\overline{SS}$ in both Master and Slave modes. Set to disable $\overline{SS}$ in both Master and Slave modes. In Slave mode, this bit has no effect if CPHA = "0".
5	MSTR	**Serial Peripheral Master** Cleared to configure the SPI as a Slave. Set to configure the SPI as a Master.
4	CPOL	**Clock Polarity** Cleared to have the SCK set to "0" in idle state. Set to have the SCK set to "1" in idle low.
3	CPHA	**Clock Phase** Cleared to have the data sampled when the SCK leaves the idle state (see CPOL). Set to have the data sampled when the SCK returns to idle state (see CPOL).
2	SPR1	SPR2 SPR1 SPR0 Serial Peripheral Rate 0 0 0 Invalid 0 0 1 $F_{CLK\ PERIPH}/4$ 0 1 0 $F_{CLK\ PERIPH}/8$ 0 1 1 $F_{CLK\ PERIPH}/16$ 1 0 0 $F_{CLK\ PERIPH}/32$ 1 0 1 $F_{CLK\ PERIPH}/64$ 1 1 0 $F_{CLK\ PERIPH}/128$ 1 1 1 Invalid
1	SPR0	

Reset Value = 0001 0100b

Figure 3: SPCON Register

7	6	5	4	3	2	1	0
SPIF	WCOL	SSERR	MODF	-	-	-	-

Bit Number	Bit Mnemonic	Description
7	SPIF	**Serial Peripheral data transfer flag** Cleared by hardware to indicate data transfer is in progress or has been approved by a clearing sequence. Set by hardware to indicate that the data transfer has been completed.
6	WCOL	**Write Collision flag** Cleared by hardware to indicate that no collision has occurred or has been approved by a clearing sequence. Set by hardware to indicate that a collision has been detected.
5	SSERR	**Synchronous Serial Slave Error flag** Set by hardware when $\overline{SS}$ is de- asserted before the end of a received data. Cleared by disabling the SPI (clearing SPEN bit in SPCON).
4	MODF	**Mode Fault** Cleared by hardware to indicate that the $\overline{SS}$ pin is at appropriate logic level, or has been approved by a clearing sequence. Set by hardware to indicate that the $\overline{SS}$ pin is at inappropriate logic level.
3	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit

Figure 4: SPSTA

2. Make CS pin of ADC low to select it.
3. As shown in timing diagram, send logic high to D_{in} of ADC to indicate start of conversion, followed by single/differential mode selection, followed by channel selection for ADC input.
4. The corresponding digital data is thus obtained at D_{out} pin - Null bit followed by 10 bits (MSB first). (Refer MCP3008 datasheet for more details.)

Table 77. SPDAT Register

SPDAT - Serial Peripheral Data Register (0C5H)

7	6	5	4	3	2	1	0
R7	R6	R5	R4	R3	R2	R1	R0

Reset Value = Indeterminate

R7:R0: Receive data bits

SPCON, SPSTA and SPDAT registers may be read and written at any time while there is no on-going exchange. However, special care should be taken when writing to them while a transmission is on-going:

Figure 5: SPDAT

TCON	88h	Timer/Counter 0 and 1 control	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
TMOD	89h	Timer/Counter 0 and 1 Modes	GATE1	C/T1#	M11	M01	GATE0	C/T0#	M10	M00

Figure 6: Timer registers

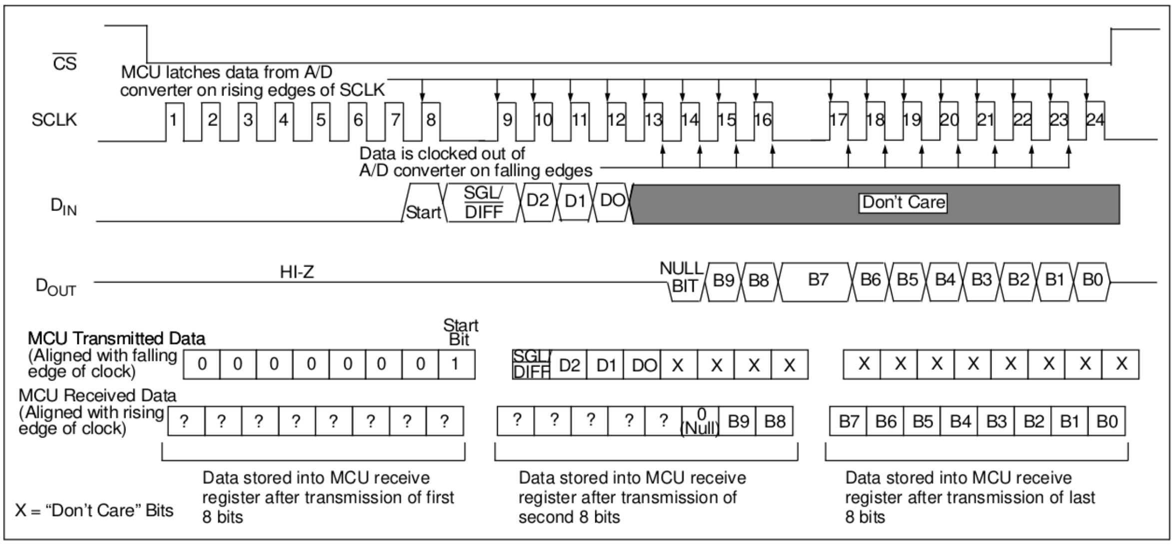

Figure 7: Timing diagram

Table 65. IEN1 Register

IEN1 - Interrupt Enable Register (B1h)

7	6	5	4	3	2	1	0
-	EUSB	-	-	-	ESPI	ETWI	EKB

Bit Number	Bit Mnemonic	Description
7	-	**Reserved**
6	EUSB	**USB Interrupt Enable bit** Cleared to disable USB interrupt. Set to enable USB interrupt.
5	-	**Reserved**
4	-	**Reserved**
3	-	**Reserved**
2	ESPI	**SPI interrupt Enable bit** Cleared to disable SPI interrupt. Set to enable SPI interrupt.
1	ETWI	**TWI interrupt Enable bit** Cleared to disable TWI interrupt. Set to enable TWI interrupt.
0	EKB	**Keyboard interrupt Enable bit** Cleared to disable keyboard interrupt. Set to enable keyboard interrupt.

Figure 8: IEN1 register

5. After the successful reception of 10 bit of data, CS pin of ADC is made high.
6. The data bits transmission and reception from microcontroller should be done as per the alignment shown in

Table 62. IEN0 Register

IEN0 - Interrupt Enable Register (A8h)

7	6	5	4	3	2	1	0
EA	EC	ET2	ES	ET1	EX1	ET0	EX0

Bit Number	Bit Mnemonic	Description
7	EA	**Enable All interrupt bit** Cleared to disable all interrupts. Set to enable all interrupts.
6	EC	**PCA interrupt enable bit** Cleared to disable. Set to enable.
5	ET2	**Timer 2 overflow interrupt Enable bit** Cleared to disable Timer 2 overflow interrupt. Set to enable Timer 2 overflow interrupt.
4	ES	**Serial port Enable bit** Cleared to disable serial port interrupt. Set to enable serial port interrupt.
3	ET1	**Timer 1 overflow interrupt Enable bit** Cleared to disable Timer 1 overflow interrupt. Set to enable Timer 1 overflow interrupt.
2	EX1	**External interrupt 1 Enable bit** Cleared to disable external interrupt 1. Set to enable external interrupt 1.
1	ET0	**Timer 0 overflow interrupt Enable bit** Cleared to disable timer 0 overflow interrupt. Set to enable timer 0 overflow interrupt.
0	EX0	**External interrupt 0 Enable bit** Cleared to disable external interrupt 0. Set to enable external interrupt 0.

Figure 9: IEN0 register

TABLE 5-2: CONFIGURE BITS FOR THE MCP3008

Control Bit Selections				Input Configuration	Channel Selection
Single /Diff	D2	D1	D0		
1	0	0	0	single-ended	CH0
1	0	0	1	single-ended	CH1
1	0	1	0	single-ended	CH2
1	0	1	1	single-ended	CH3
1	1	0	0	single-ended	CH4
1	1	0	1	single-ended	CH5
1	1	1	0	single-ended	CH6
1	1	1	1	single-ended	CH7
0	0	0	0	differential	CH0 = IN+ CH1 = IN-
0	0	0	1	differential	CH0 = IN- CH1 = IN+
0	0	1	0	differential	CH2 = IN+ CH3 = IN-
0	0	1	1	differential	CH2 = IN- CH3 = IN+
0	1	0	0	differential	CH4 = IN+ CH5 = IN-
0	1	0	1	differential	CH4 = IN- CH5 = IN+
0	1	1	0	differential	CH6 = IN+ CH7 = IN-
0	1	1	1	differential	CH6 = IN- CH7 = IN+

Figure 10: MCP 3008 configure bits

timing diagram.

4 Voltage conversion formulae

$$\text{LSB size} = \frac{V_{ref}}{1024}$$
$$\text{Digital output code produced by the ADC} = \frac{1024 \times V_{in}}{V_{ref}}$$

where, V_{ref} - Reference voltage, V_{in} - Analog input voltage

5 Temperature sensor - LM35

The reference voltage provided to the ADC is 3.3V and the ADC has a resolution of 10 bits. Hence the minimum voltage that can be measured by the ADC is only around 3.3 mV (3.3V/1024). The output of the LM35 is $10mV/^0C$.