

	Institut Supérieur des Etudes Technologiques de Sousse Département Génie Électrique <b>Examen de contrôle</b>	2020/2021 Semestre : 1 Date : Janvier 2021
	Matière : Mise à niveau Systèmes embarqués	Classes : Master EASER – M1
Documents : Autorisés	Enseignant : Ali HMIDENE	Nb. Pages : 10 pages

## Problème

Le schéma de la figure 1 présente l'interfaçage d'un microcontrôleur STM32F407 avec trois circuits :

- Capteur de pression "MPX2100"
- Capteur de température "TC72"

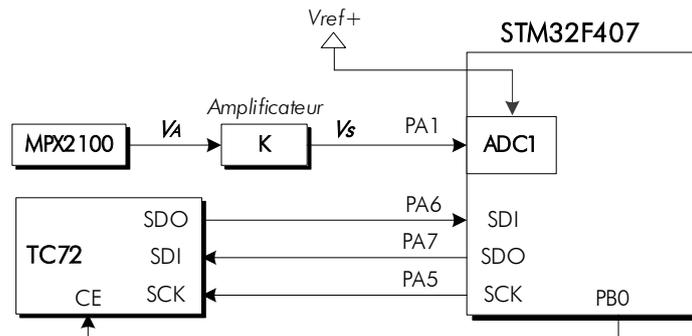


Figure 1

## PARTIE 1 – MPX2100

La série MPX2100 fournit des capteurs de pression piézorésistifs en silicium fournissant une sortie de tension linéaire et très précise directement proportionnelle à la pression appliquée. Ces capteurs délivrent des tensions entre de 0 à 40mV pour des pressions allant de 0 100kPa (ou de 0 à 1 bar).

La tension de sortie  $V_A$  est amplifiée par un amplificateur de Gain  $K$  et appliquée à l'entrée PA1 (Canal 1) de l'ADC1 du microcontrôleur STM32F407.

Pour  $V_{ref+} = 1,5V$  et  $K = 30$

1. Donner la valeur minimale et maximale de la tension à la sortie de l'amplificateur.

$$V_{Smin} = 0V, V_{Smax} = 40 \cdot 10^{-3} \times 30 = 1,2V$$

2. Donner la résolution du convertisseur Analogique numérique pour garantir une précision inférieure de 1mbar.

1mbar représente le 1/1000 de la pleine échelle. Cette valeur correspond à 1,2mV. La résolution du convertisseur =  $\frac{1,5V}{1,2mV} = 1250$  niveaux. Ce qui correspond à une résolution de 11 bits

3. Calculer le temps d'acquisition  $T_S$  (en nombre de cycles) pour les paramètres suivants :

$$R_{AIN} = \text{négligeable}; R_{ADC} = 2k\Omega; C_{ADC} = 4pF; F_{ADC} = 25Mhz \text{ et } N = 12bits;$$

$$T_S = R_{ADC} \times F_{ADC} \times C_{ADC} \times \ln(2^{N+1}) = 2 \cdot 10^3 \times 4 \cdot 10^{-12} \times 25 \cdot 10^6 \times \ln(2^{13}) = 1,8$$

4. Compléter le pseudocode suivant pour de configurer de la broche PA1 en analogique.

```
GPIO_InitStruct.Pin = GPIO_PIN_1;
GPIO_InitStruct.Mode = GPIO_MODE_ANALOG;
GPIO_InitStruct.Pull = GPIO_NOPULL;
HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
```

5. Compléter le code de la procédure `void Init_ADC1()` pour configurer le convertisseur ADC1 en mode simple conversion sur un seul canal.

```
void Init_ADC1 (void)
```

```

{
  ADC_ChannelConfTypeDef sConfig = {0};
  hadc1.Instance = ADC1;
  hadc1.Init.ClockPrescaler = ADC_CLOCK_SYNC_PCLK_DIV2;
  hadc1.Init.Resolution = ADC_RESOLUTION_12B;
  hadc1.Init.ScanConvMode = DISABLE;
  hadc1.Init.ContinuousConvMode = DISABLE;
  hadc1.Init.DiscontinuousConvMode = DISABLE;
  hadc1.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_NONE;
  hadc1.Init.ExternalTrigConv = ADC_SOFTWARE_START;
  hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;
  hadc1.Init.NbrOfConversion = 1;
  hadc1.Init.EOCSelection = ADC_EOC_SINGLE_CONV;
  HAL_ADC_Init(&hadc1) ;

  sConfig.Channel = ADC_CHANNEL_1;
  sConfig.Rank = 1;
  sConfig.SamplingTime = ADC_SAMPLETIME_3CYCLES;
  HAL_ADC_ConfigChannel(&hadc1, &sConfig) ;
}

```

6. Ecrire le code de la fonction `uint32_t ReadADC(void)` qui renvoi en sortie la valeur convertie.

```

uint32_t ReadADC(void){
  uint32_t ADC_Val;
  HAL_ADC_Start(&hadc1);
  HAL_ADC_PollForConversion(&hadc1, 1) ;
  ADC_Val = HAL_ADC_GetValue(&hadc) ;
  return(ADC_Val) ;
}

```

## PARTIE 2 – TC72

---

Le TC72 est un capteur de température numérique, qui peut être connecté au bus SPI via les lignes SDI, SDO et SCL. La ligne CE, active au niveau haut, est utilisée particulièrement, lorsqu'on veut connecter plusieurs circuits sur le bus.

Le TC72 peut opérer en mode "One Shot" ou en mode continu. En mode One Shot, la mesure de la température s'effectue après une demande de lecture. Alors, qu'en mode continu, la mesure est effectuée approximativement toutes les 150ms.

Le circuit TC72 peut mesurer des températures de  $-55^{\circ}\text{C}$  à  $+125^{\circ}\text{C}$ . La valeur numérique est codée sur 10 bits en complément à 2 avec une précision de  $0,25^{\circ}\text{C}/\text{bit}$ . Cette valeur est représentée sur deux registres de 8 bits. Le registre MSB contient la partie entière et le registre LSB contient la partie fractionnaire. Uniquement les bits 7 et 6 sont utilisés dans ce dernier registre.

D7	D6	D5	D4	D3	D2	D1	D0	Register
Sign	$2^6$	$2^5$	$2^4$	$2^3$	$2^3$	$2^1$	$2^0$	Temp. MSB
$2^{-1}$	$2^{-2}$	0	0	0	0	0	0	Temp. LSB

## Opération de lecture/écriture

Le TC72 peut opérer sur le front montant ou descendant du signal d'horloge SCK. La polarité de l'horloge est déterminée lors du passage de la ligne CE à l'état haut. Nous nous plaçons dans ce problème dans le cas où CPOL = 0 et CPHA = 1. La fréquence maximale de transfert est de 7,5MHz.

Le transfert de donnée consiste à envoyer en premier lieu un octet d'adresse suivie d'un ou de plusieurs octets de données. Le bit MSB d'adresse (A7) détermine si une opération de lecture ou écriture aura lieu. Si A7 = 1, un ou plusieurs cycles d'écriture sont requis ; si A7 = 0, une opération de lecture est requis. La TC72 dispose de quatre registres, un registre de contrôle deux registres de données et un registre ID (un code pour l'identification du circuit).

Register	Read Address	Write Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR/BOR
Control	00hex	80hex	0	0	0	One-Shot	0	1	0	Shutdown	05hex
LSB Temperature	01hex	-	T1	T0	0	0	0	0	0	0	00hex
MSB Temperature	02hex	-	T9	T8	T7	T6	T5	T4	T3	T2	00hex
Manufacturer ID	03hex	-	0	1	0	1	0	1	0	0	54hex

Le registre de contrôle permet à travers les bits **One Shot** et **Shutdown** de sélectionner le mode Shutdown (mode économique), conversion continue ou à la demande (One Shot).

Operational Mode	One-Shot Bit 4	Shutdown Bit 0
Continuous Temperature Conversion	0	0
Shutdown	0	1
Continuous Temperature Conversion (One-Shot Command is ignored if Shutdown = '0')	1	0
One-Shot	1	1

A chaque fois que la combinaison One-Shot = 1 et Shutdown = 1 est sélectionné (mode One Shot), une seule mesure est effectuée. Après la fin de conversion le bit **One-Shot** revient à zéro (le circuit revient au mode Shutdown).

Que ce soit en mode One Shot ou continu, la donnée de mesure devient disponible après 150ms environ.

La figure 2 illustre les chronogrammes des opérations d'écriture et de lecture.

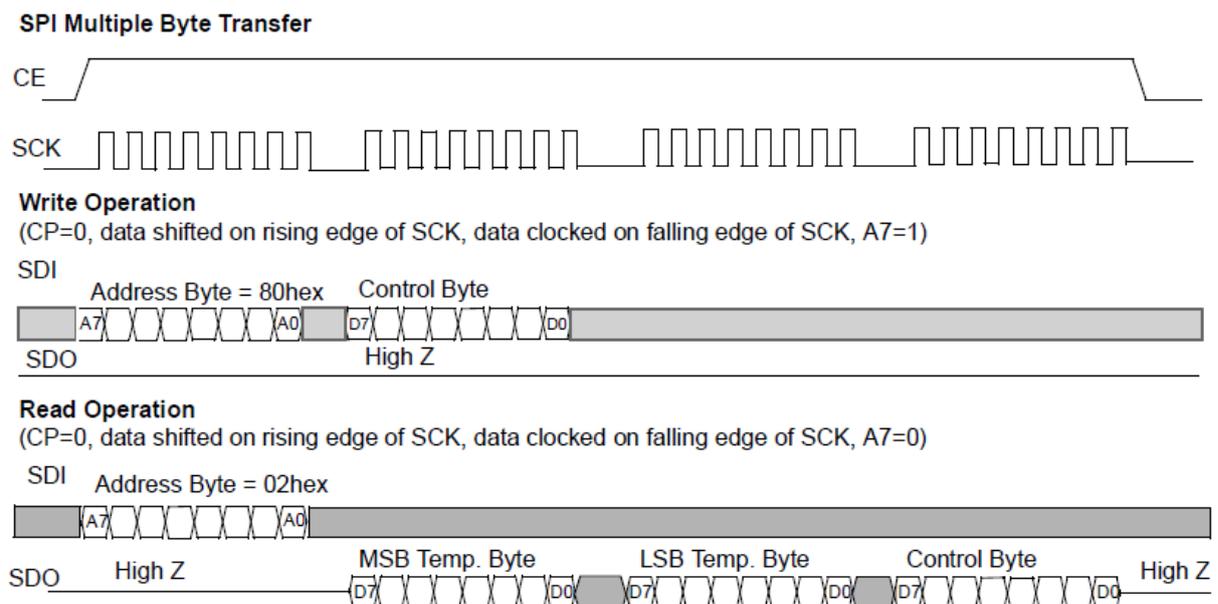


Figure 2

## Programmation

7. Compléter le pseudocode suivant pour la configuration du module SPI1. La fréquence du bus APB2 étant égale à 50MHz.

```
hspi1.Instance = SPI1;
hspi1.Init.Mode = SPI_MODE_MASTER;
hspi1.Init.Direction = SPI_DIRECTION_2LINES;
hspi1.Init.DataSize = SPI_DATASIZE_8BIT;
hspi1.Init.CLKPolarity = SPI_POLARITY_LOW;
hspi1.Init.CLKPhase = SPI_PHASE_2EDGE;
hspi1.Init.NSS = SPI_NSS_SOFT;
hspi1.Init.BaudRatePrescaler = SPI_BAUDRATEPRESCALER_64;
hspi1.Init.FirstBit = SPI_FIRSTBIT_MSB;
HAL_SPI_Init(&hspi1) ;
```

La lecture de la température nécessite les étapes suivantes :

- Activez TC72 (CE = 1),
- Envoyez l'adresse 0x80 (A7 = 1), suivie de la commande One-Shot (contrôle = 0001 0101),
- Désactiver TC72 (CE = 0),
- Attendez au moins 150 ms pour que la température soit disponible,
- Activez TC72 (CE = 1, pour le transfert de données multiples),
- Envoyer la commande de lecture (adresse de lecture = 0x02),
- Lire la température deux octets consécutifs MSB puis LSB,
- Désactivez le transfert de données TC72 (CE = 0),

8. Ecrire la procédure **void Lire\_Temperature(void)** en suivant les étapes ci-dessus.

```
char TxCmd[2] ;
char RxData[2] ;
...
void Lire_Temperature(void){
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, GPIO_PIN_SET);
TxCmd[0] = 0x80; TxCmd[1] = 0x11 ;
HAL_SPI_Transmit(&hspi1, (uint8_t *)TxCmd, 2, 200);
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, GPIO_PIN_RESET);
HAL_Delay(200);
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, GPIO_PIN_SET);
TxCmd[0] = 0x02;
HAL_SPI_Transmit(&hspi1, (uint8_t *)TxCmd, 1, 200);
HAL_SPI_Receive(&hspi1, (uint8_t *)&RxData, 2, 200);
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, GPIO_PIN_RESET);
}
```