	Institut Supérieur des Etudes Technologiques de Sousse **Département Génie Electrique** Examen de contrôle	***Année universitaire*** *: 2015/2016* ***Semestre*** *: 2* ***Date :*** *21 Juin 2016*
Matière *: DSC*	***Classes*** *: Master EASER – M1*	***Durée :*** *1h30mn*
Documents *: Non autorisés*	***Enseignant*** *: Ali HMIDENE*	***Nb. Pages*** *: 10 pages*

Problème

Les panneaux photovoltaïques assurent une production décentralisée pour alimenter des matériels portatifs ou satisfaire des besoins locaux en des lieux isolés, mais ils participent aussi à la politique énergétique globale grâce à leur connexion aux réseaux de distribution d'électricité.

Pour accroitre le rendement du système photovoltaïque, il est possible d'utiliser des panneaux solaires capables de suivre le soleil, ce qui permet de récupérer constamment le maximum d'énergie lumineuse. Suivre le soleil permet une amélioration de rendement de l'ordre de 40% par rapport à un système fixe.

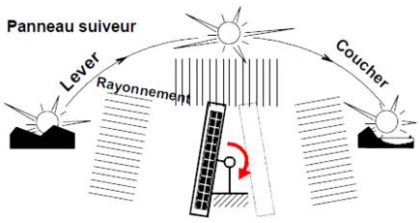

Figure 1

En un point de la surface terrestre, le rayonnement solaire est difficile à capter car la position apparente du soleil dans la voûte céleste ne cesse de changer. Les mouvements relatifs de la Terre par rapport au soleil sont décrits par les lois complexes de l'Astronomie.

Pour un observateur situé en un lieu précis de la surface terrestre (latitude + longitude), la position du soleil dans le ciel, à un instant donné, peut être repérée par 2 coordonnées angulaires :

- L'AZIMUT : angle mesuré par rapport au sud dans un plan horizontal,
- L'ELEVATION : angle mesuré par rapport à l'horizontale dans un plan vertical.

Pointage optimal du panneau :

Pour que la production photovoltaïque soit maximale, les rayons provenant directement du soleil doivent avoir un angle d'incidence égal à 90°.

Le pointage du panneau est donc optimal lorsque la normale au plan du panneau, en son centre, est dirigée vers le soleil.

Les 2 capteurs solaires ne sont pas fixes par rapport au sol, mais montés sur le cadre du panneau solaire et donc liés à ses mouvements.

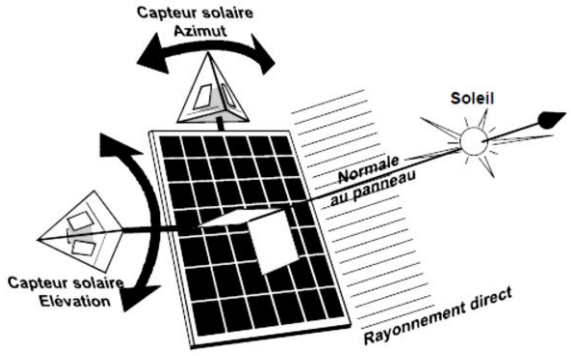

Figure 2

En fonction du déplacement apparent du soleil sur sa trajectoire, la phase du suivi se résume à augmenter ou à diminuer l'azimut et/ou l'élévation du panneau solaire (figure 2).

On se propose dans ce sujet de faire la commande simplifiée d'un suiveur de soleil. On s'intéresse ***uniquement à l'augmentation et la diminution de l'Azimut***. Le principe utilisé repose sur l'exploitation d'un capteur « DegerConecter ».

Le DegerConecter présente la forme d'une pyramide à base triangulaire et constitue un capteur capable de percevoir la présence et la position relative d'une source lumineuse (Figure 3). Le capteur est composé de 3 cellules sensibles à l'éclairement solaire. Pendant la phase de suivi solaire, les cellules n°1 et n°2 sont constamment, mais différemment, exposées aux rayons du soleil. En effet, quand le soleil se déplace, il apparaît un angle relatif α entre la direction des rayons du soleil et la normale au panneau, ce qui provoque un éclairement différent des cellules n°1 et n°2, situées de part et d'autre de l'arête frontale du capteur (Figure 4). La cellule n°3 orientée vers l'arrière du panneau reste « dans l'ombre » sans être directement éclairée.

Le schéma de connexion des cellules est donné par la figure 5. La tension V_A aux bornes de la résistance R_A $(V_A = k \cdot \alpha = V_1 - V_2)$ est amplifiée $V_S = V_A + \frac{V_{ref+}}{2}$ puis traitée par une carte électronique à base d'un microcontrôleur STM32F207 afin de commander le moteur du vérin (Figure 6).

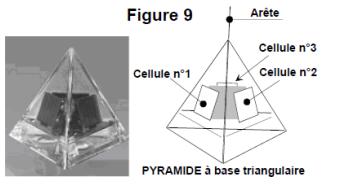

Figure 3

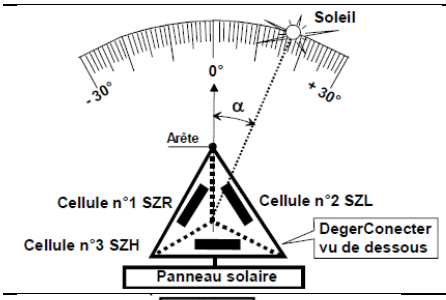

Figure 4

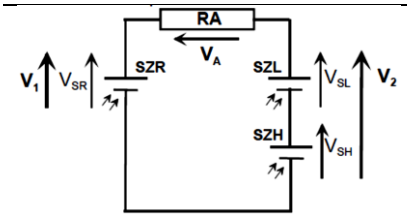

Figure 5

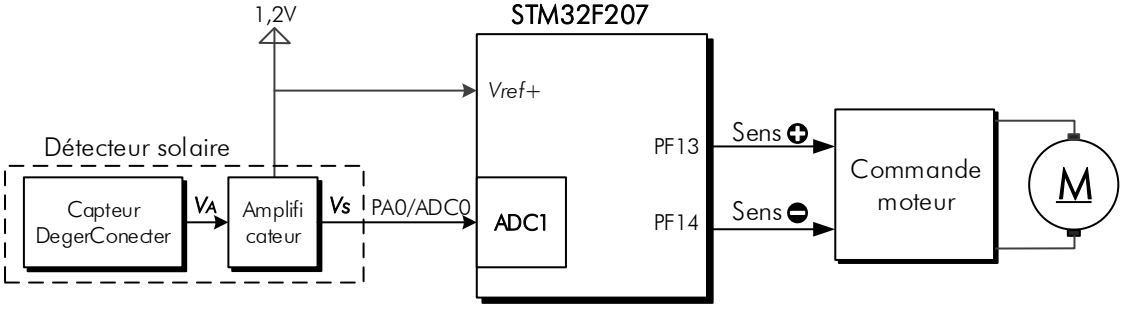

Figure 6

A tout instant, la carte électronique contenue dans le détecteur solaire amplifie, la tension VA = V1 – V2 qui est fonction de l'angle α. La tension de sortie est appliquée à l'entrée du convertisseur analogique numérique ADC1. Le microcontrôleur commande le moteur (dans le sens adéquat) de façon à rejoindre la position $\alpha = 0^\circ$. Si $V_S > \frac{V_{ref+}}{2}$, cela signifie que le suiveur est « en retard » par rapport au soleil, le suiveur est commandé dans le « Sens +» afin de compenser ce retard jusqu'à ce que la tension $V_S = \frac{V_{ref+}}{2}$. Si $V_S < \frac{V_{ref+}}{2}$ cela signifie que le suiveur est « Avance » par rapport au soleil, le suiveur est commandé dans le « Sens — » jusqu'à ce que la tension $V_S = \frac{V_{ref+}}{2}$.

PARTIE 1 – GPIO –

Les sorties logiques sont connectées au microcontrôleur selon le schéma de la figure 6.

1. Quels sont les niveaux logiques à fournir aux sorties PF13 et PF14 dans les cas suivants :
   a. Moteur tourne dans le « Sens+ ».
   b. Moteur tourne dans le « Sens — ».
   c. Moteur à l'arrêt
2. Ecrire le pseudocode de configuration des lignes PF13 et PF14.

PARTIE – ADC –

La sortie du détecteur solaire (Vs) est connectée à l'entrée PA0 (canal 0) du convertisseur analogique numérique ADC1.

3. Sur combien de bits doit-on coder le signal à convertir pour garantir un pas de quantification inférieur à 1,5mV ?
4. Pour les paramètres suivants :
   - $R_{AIN} = 5K\Omega$
   - $R_{ADC} = 2K\Omega$
   - $C_{ADC} = 4pf$
   - $F_{ADC} = 15MHz$
   - Fréquence du bus APB2 = 60MHz.

   Calculer le temps d'acquisition Ts.
5. Ecrire le pseudocode de la configuration de la broche PA0 en analogique.
6. Ecrire le code de la procédure Init_ADC1( ) qui initialise le convertisseur ADC1 en mode simple conversion sur un seul canal ($F_{ADC} = 15Mhz$ et $T_S = 3cycles$).
7. Ecrire le pseudocode de l'acquisition du signal provenant du détecteur solaire et de sauvegarder la valeur convertie dans la variable globale Val_Azimut.
8. Ecrire le code de la procédure Commande( ), qui commande le moteur selon les valeurs de la variable Val_Azimut.

ANNEXES

```
typedef struct
{
  __IO uint32_t MODER;   /* GPIO port mode register,              Address offset: 0x00*/
  __IO uint32_t OTYPER;  /* GPIO port output type register,       Address offset: 0x04*/
  __IO uint32_t OSPEEDR; /* GPIO port output speed register,      Address offset: 0x08*/
  __IO uint32_t PUPDR;   /* GPIO port pull-up/pull-down register, Address offset: 0x0C*/
  __IO uint32_t IDR;     /* GPIO port input data register,        Address offset: 0x10*/
  __IO uint32_t ODR;     /* GPIO port output data register,       Address offset: 0x14*/
  __IO uint16_t BSRRL;   /* GPIO port bit set/reset low register, Address offset: 0x18*/
  __IO uint16_t BSRRH;   /* GPIO port bit set/reset high register,Address offset: 0x1A*/
  __IO uint32_t LCKR;    /* GPIO port configuration lock register,Address offset: 0x1C*/
  __IO uint32_t AFR[2]; /* GPIO alternate function registers,Address offset: 0x24-0x28*/
} GPIO_TypeDef;

#define GPIOA                ((GPIO_TypeDef *) GPIOA_BASE)
#define GPIOB                ((GPIO_TypeDef *) GPIOB_BASE)
#define GPIOC                ((GPIO_TypeDef *) GPIOC_BASE)
#define GPIOD                ((GPIO_TypeDef *) GPIOD_BASE)
#define GPIOE                ((GPIO_TypeDef *) GPIOE_BASE)
#define GPIOF                ((GPIO_TypeDef *) GPIOF_BASE)
#define GPIOG                ((GPIO_TypeDef *) GPIOG_BASE)
#define GPIOH                ((GPIO_TypeDef *) GPIOH_BASE)
#define GPIOI                ((GPIO_TypeDef *) GPIOI_BASE)
```

Configuration des GPIO

MODER(i) [1:0]	OTYPER(i)	OSPEEDR(i) [B:A]	PUPDR(i) [1:0]		Configuration des I/O	
01	0	SPEED [B:A]	0	0	GP output	PP
	0		0	1	GP output	PP + PU
	0		1	0	GP output	PP + PD
	0		1	1	Reservé	
	1		0	0	GP output	OD
	1		0	1	GP output	OD + PU

	1			1	0	GP output	OD + PD
	1			1	1	Reservé (GP output OD)	
10	0	SPEED [B:A]		0	0	AF	PP
	0			0	1	AF	PP + PU
	0			1	0	AF	PP + PD
	0			1	1	Reservé	
	1			0	0	AF	OD
	1			0	1	AF	OD + PU
	1			1	0	AF	OD + PD
	1			1	1	Reservé	
00	x	x	x	0	0	input	Floating
	x	x	x	0	1	input	PU
	x	x	x	1	0	input	PD
	x	x	x	1	1	Reservé (input floating)	
11	x	x	x	0	0	Input / output	Analog
	x	x	x	0	1	Reservé	
	x	x	x	1	0		
	x	x	x	1	1		

GP = general-purpose, PP = push-pull, PU = pull-up, PD = pull-down, OD = open-drain, AF = alternate function

GPIO port mode register (GPIOx_MODER) (x = A..I)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

MODERy[1:0] : configuration Port x bits (y = 0..15)
- 00 : Input (reset state)
- 01 : General purpose output mode
- 10 : Alternate function mode
- 11 : Analog mode

Après un Reset :
- 0xA800 0000 pour port A
- 0x0000 0280 pour port B
- 0x0000 0000 pour les autres ports

GPIO port output type register (GPIOx_OTYPER) (x = A..I)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

OTy : Port x configuration bits (y = 0..15)
- 0 : Output push-pull (reset state)
- 1 : Output open-drain

Après un Reset :
- 0x0000 00C0 pour port B
- 0x0000 0000 pour les autres ports

GPIO port output speed register (GPIOx_OSPEEDR) (x = A..I)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEEDR15 [1:0]		OSPEEDR14 [1:0]		OSPEEDR13 [1:0]		OSPEEDR12 [1:0]		OSPEEDR11 [1:0]		OSPEEDR10 [1:0]		OSPEEDR9 [1:0]		OSPEEDR8 [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSPEEDR7[1:0]		OSPEEDR6[1:0]		OSPEEDR5[1:0]		OSPEEDR4[1:0]		OSPEEDR3[1:0]		OSPEEDR2[1:0]		OSPEEDR1 [1:0]		OSPEEDR0 1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

OSPEEDRy[1:0] : Port x configuration bits (y = 0..15)

00 : Low speed
01 : Medium speed
10 : Fast speed
11 : High speed

Après un Reset :
- 0x0000 00C0 pour port B
- 0x0000 0000 pour les autres ports

GPIO port pull-up/pull-down register (GPIOx_PUPDR) (x = A..I)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

PUPDRy[1:0] : Port x configuration bits (y = 0..15)

00 : No pull-up, pull-down
01 : Pull-up
10 : Pull-down
11 : Reserved

Après un Reset :
- 0x6400 0000 pour port A
- 0x0000 0100 pour port B
- 0x0000 0000 pour les autres ports

GPIO port input data register (GPIOx_IDR) (x = A..I)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

IDRy : Port input data (y = 0..15)

Après un Reset :
- 0x0000 XXXX (X indéfinie)

GPIO port output data register (GPIOx_ODR) (x = A..I)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ODRy: Port output data (y = 0..15)

Après un Reset :
- 0x0000 0000

GPIO port bit set/reset register (GPIOx_BSRR) (x = A..I)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

These bits are write-only and can be accessed in word, half-word or byte mode.
A read to these bits returns the value 0x0000.

BRy: Port x reset bit y (y = 0..15)

0 : No action on the corresponding ODRx bit
1 : Resets the corresponding ODRx bit

Note: If both BSx and BRx are set, BSx has priority.

BSy: Port x set bit y (y= 0..15)

0 : No action on the corresponding ODRx bit
1 : Sets the corresponding ODRx bit

Après un Reset :
- 0x0000 0000