

Les sorties PWM

Introduction

Le PWM – **P**ulse **W**idth **M**odulation ou modulation de largeur d'impulsions, permet de créer un signal digital (niveaux bas et haut) dont le rapport cyclique est paramétrable.

Certain microcontrôleur dispose de circuits spécialités permettant la génération de signaux PWM: c'est à dire qu'une fois paramétrée, la sortie génère le signal sans l'intervention du µC.

La carte Spider dispose de 6 canaux PWM.

Attention



Les sorties PWM partagent les mêmes horloges internes (timer). Cela signifie que le changement de fréquence sur une broche affectera également d'autres sorties utilisant le même timer. Toutefois, chaque sortie PWM utilise un compteur indépendant pour générant la durée de l'impulsions ou du rapport cyclique.

Voir tableau plus bas pour la répartition des timers et des sorties.

Emplacement des sorties PWM

Brochage de connecteurs Gadgeteer de type P

Socket	Pin1	Pin2	Pin3	Pin4	Pin5	Pin6	Pin7	Pin8	Pin9	Pin10
P	+3.3V	+5V	GPIO!	[UN]	[UN]	GPIO	PWM (G)	PWM (G)	PWM	GND

Canaux PWM des cartes Spider I et II:

Type Socle	N° socle	N° broche	Spider II		Spider I	
			Canaux	N° port	Canaux	N port
P	8	7	PWM11 (Timer 2)	P3.29	PWM5 (Timer 1)	IO74
		8	PWM9 (Timer 2)	P3.27	PWM4 (Timer 1)	IO48
		9	PWM8 (Timer 2)	P3.26	PWM3 (Timer 1)	IO49
	11	7	PWM6 (Timer 2)	P3.24	PWM1 (Timer 1)	IO14
		8	PWM0 (Timer 1)	P3.16	PWM0 (Timer 0)	IO13
		9	PWM1 (Timer 1)	P3.17	PWM2 (Timer 0)	IO50

Déclaration et initialisation

La fonction PWM est implémentée par l'*assembly* **Microsoft.SPOT.Hardware.PWM**

L'espace de nom est: **Microsoft.SPOT.Hardware**

```
using System;
using Microsoft.SPOT;
using Microsoft.SPOT.Hardware;
public class Program
{
    public static void Main()
    {
        bool invertedOutput= false;
        double frequence = 10000; // Période en microseconde
        double rapportCyclique = 0.10; // Période en microseconde
        PWM LED = new PWM((Cpu.PWMChannel)9, frequence, rapportCyclique, invertedOutput);
        LED.Start();
    }
}
```

Enumération

Dans NETMF, les broches PWM sont définies dans l'énumération "Cpu.PWMChannel". Le nombre de broches PWM peut dépasser le nombre de membres contenu dans "Cpu.PWMChannel". L'emploi d'un cast permet d'utiliser des s PWMs supplémentaires.

```
// Utilisation de l'énumération
PWM MyServo = new PWM(Cpu.PWMChannel.PWM_3, 2175, 175,
    PWM.ScaleFactor.Microseconds, false);

// Pour les PWMs non énumérés, utiliser un cast pour accéder un PWM supplémentaire
PWM LED = new PWM((Cpu.PWMChannel)9, 10000, 0.10, false);
```



Cast

La Classe PWM

Définie dans l'assembly **Microsoft.SPOT.Hardware.PWM** **NETMF 4.3**

Espace de nom: **Microsoft.SPOT.Hardware**

Constructeur 1 – fréquence et rapport cyclique

```
Public PWM (
    Cpu.PWMChannel channel,
    double frequency_Hz,
    double dutyCycle,
    bool invert
)
```

Initialise une nouvelle instance de la classe PWM avec le canal spécifié, la fréquence, le rapport cyclique et le flag d'inversion de la sortie.

Paramètres

Channel : **Microsoft.SPOT.Hardware.Cpu.PWMChannel**

Canal du port PWM.

frequency_Hz Type: **System.Double**

Fréquence des impulsions.

dutyCycle Type: **System.Double**

Rapport cyclique des impulsions en fraction d'unité.
Ex : 0,5 = 50%.

Invert Type: **System.Boolean**

Flag d'inversion de la sortie.

Constructeur 2 – période et durée

```
Public PWM (
    Cpu.PWMChannel channel,
    unsigned int period,
    unsigned int duration,
    PWM.ScaleFactor scale,
    bool invert
)
```

Initialise une nouvelle instance de la classe PWM avec le canal spécifié, la période, la durée de l'impulsion et le flag d'inversion de la sortie.

Paramètres

channel **Microsoft.SPOT.Hardware.Cpu.PWMChannel**

Canal du port PWM.

period **System.UInt32**

Période de l'impulsion, exprimée selon le facteur d'échelle.

duration Type: **System.UInt32**

Durée de l'impulsion, exprimée selon le facteur d'échelle.

Scale Type: **Microsoft.SPOT.Hardware.PWM.ScaleFactor**

Facteur d'échelle de la période et de la durée, en:
Milliseconds ; Microseconds ; Nanoseconds.

Invert Type: **System.Boolean**

Flag d'inversion de la sortie.

Propriétés

public UInt32 PWM.Duration { get; set; }	Durée des impulsions du signal
public double PWM.DutyCycle { get; set; }	Rapport cyclique du signal
public double PWM.Frequency { get; set; }	Fréquence des impulsions
public UInt32 PWM.Period { get; set; }	Période du signal.
public Pin PWM.Pin { get; }	Retourne le n° de la broche du canal PWM.
public PWM.ScaleFactor Scale { get; set; }	Facteur d'échelle pour la durée et la période.

Méthodes

```
public void Start () // Démarre le port PWM pour une durée indéfinie.  
public static void Start ( PWM[] ports ) // Lance un certain nombre de ports PWM en même temps.  
Parameters  
Ports:tType: array<Microsoft.SPOT.Hardware.PWM The ports to start.  
  
public void Stop () // Arrête le port PWM.  
public static void Stop ( PWM [ ] ports ) // Arrête un certain nombre de ports PWM en même temps.
```