

1.	Verslag	2
1.1	Testconfiguratie klauw via Codesys	2
1.1.1	PWM0CTRL0 – berekening basisfrequentie:.....	3
1.1.2	PWM0CTRL1.....	4
1.1.3	PWM0CHA	4
1.2	Webfunctionaliteit	4

1. Verslag

1.1 Testconfiguratie klauw via Codesys

Via Codesys wordt nu op een softwarematige manier de frequentie en de duty cycle geprogrammeerd en wordt deze getest via online besturing in de browser.

Om de minimum- en maximumwaarden van de servomotoren te kunnen begrenzen moeten eerst de nodige berekeningen worden uitgevoerd.

De PLC-module heeft vier PWM-functies:

- **Servo-mode:** $f=50\text{Hz}$, 1ms minimum, 2ms maximum.
- **Duty cycle mode:** de frequentie wordt berekend voor beide kanalen en de duty cycle wordt apart berekend voor elk kanaal.
- **Universal mode:** zelfde als duty cycle, maar de duty cycle van het tweede kanaal is niet configureerbaar en blijft op 50%.
- **Frequency mode:** frequenties zijn regelbaar tot 20kHz voor beide kanalen, maar de duty cycle blijft op 50%.

Voor het project wordt duty cycle mode gebruikt aangezien altijd op 50Hz wordt gewerkt en de minimum tijdswaarde ongeveer 0,5ms bedraagt en de maximumwaarden tot 2,5ms lopen.

De mode wordt via Codesys aan de PLC doorgegeven onder de vorm van bytes en 16bit woorden via mapping aan de geconfigureerde adressen.

Voor de PLC zijn reeds aan alle functies een adres toegekend. Deze configuratiefile is te downloaden op de website van Codesys. De ontwikkelomgeving is gratis, maar de functies van de PLC's dienen aangekocht te worden. Industriële PLC-omgevingen zijn behoorlijk duur, maar deze voor Raspberry Pi was te verkrijgen voor 90€. Hiervoor krijg je quadcore-support, volledige support voor de PLC qua functionaliteit en Softmotion. Softmotion is complexe software ontwikkeld voor robotica en CNC toepassingen. In dit project wordt geen gebruik gemaakt van Softmotion aangezien de functieblokken onder andere complexe berekeningen vereisen op gebied van statica, dynamica en regeltechniek.

Er wordt vooreerst een lijst met variabelen aangemaakt:

Scope	Name	Address	Data type	Initialization	Comment	Attributes
VAR_GLOBAL	GPIO24MASTER		BOOL			
VAR_GLOBAL	PWM0CTRL1		WORD	2500		
VAR_GLOBAL	AI0	%IW100	WORD	71		
VAR_GLOBAL	PWM0CHA		WORD	50		
VAR_GLOBAL	PWMCTRL0		BYTE	105		

Deze worden toegewezen aan de functies van de PLC.

Application.GVL.PW...	PWM0Ctrl1	%QW9	WORD	The effect of the PWM0Ctrl1 value depends on the selected Mode in PWM0Ctrl0. This value applies to channels A & B of PWM
	PWM1Ctrl1	%QW1	WORD	The effect of the PWM1Ctrl1 value depends on the selected Mode in PWM1Ctrl0. This value applies to channels A & B of PWM
	PWM2Ctrl1	%QW2	WORD	The effect of the PWM2Ctrl1 value depends on the selected Mode in PWM2Ctrl0. This value applies to channels A & B of PWM
Application.GVL.PW...	PWM0Ctrl0	%QB6	BYTE	PWM 0 Control 0 - Allows to set the PWM mode, channel enable and prescaler. PWM 0 uses 16 Bit values.
	PWM1Ctrl0	%QB7	BYTE	PWM 1 Control 0 - Allows to set the PWM mode, channel enable and prescaler. PWM 1 uses 16 Bit values.
	PWM2Ctrl0	%QB8	BYTE	PWM 2 Control 0 - Allows to set the PWM mode, channel enable and prescaler. PWM 2 uses 16 Bit values.
	GPIODebounce01	%QB9	BYTE	Debounce the GPIO Inputs 0 and 1, value * Bus-Cycle interval = debounce time.
	GPIODebounce23	%QB10	BYTE	Debounce the GPIO Inputs 2 and 3, value * Bus-Cycle interval = debounce time.
	DigitalInDebounce01	%QB11	BYTE	Debounce the Digital Inputs 0 and 1, value * Bus-Cycle interval = debounce time.
	DigitalInDebounce23	%QB12	BYTE	Debounce the Digital Inputs 2 and 3, value * Bus-Cycle interval = debounce time.
	DigitalInDebounce45	%QB13	BYTE	Debounce the Digital Inputs 4 and 5, value * Bus-Cycle interval = debounce time.
	DigitalInDebounce67	%QB14	BYTE	Debounce the Digital Inputs 6 and 7, value * Bus-Cycle interval = debounce time.
	DigitalInDebounce89	%QB15	BYTE	Debounce the Digital Inputs 8 and 9, value * Bus-Cycle interval = debounce time.
	DigitalInDebounce1011	%QB16	BYTE	Debounce the Digital Inputs 10 and 11, value * Bus-Cycle interval = debounce time.
	DigitalInDebounce1213	%QB17	BYTE	Debounce the Digital Inputs 12 and 13, value * Bus-Cycle interval = debounce time.
	DigitalInDebounce1415	%QB18	BYTE	Debounce the Digital Inputs 14 and 15, value * Bus-Cycle interval = debounce time.
	SafeState	%QX19.0	BIT	Switch microcontroller to a safe state, in case the PLC needs to reboot or shutdown. True = Go to Safe State, False = Stay O
	RetainDataEnable	%QX19.1	BIT	Activate the retain data storage function in the microcontroller. TRUE = On, FALSE = Off.
	RetainCopy	%QX19.2	BIT	If TRUE then the current RetainDataOut is copied to RetainDataIn within the microcontroller. If FALSE the currently stored da
State				
Analog Inputs				
Digital Inputs				
Digital Outputs				
PWM Outputs				
Application.GVL.PW...	PWM0A	%QW12	WORD	PWM 0, channel A output value, the actual PWM behaviour depends on the mode setting in PWM0CtrlLH.
	PWM0B	%QW13	WORD	PWM 0, channel B output value, the actual PWM behaviour depends on the mode setting in PWM0CtrlLH.

1.1.1 PWM0CTRL0 – berekening basisfrequentie:

Bit	7	6	5	4	3	2	1	0
Name	PS2	PS1	PS0	EnableB	EnableA	-	MODE1	MODE0
Startvalue	0	0	0	0	0	0	0	0
MODE1	MODE0	Mode-Name		Description				
0	0	Servo-Mode		Both channels (A / B) are used to drive model-building servos - special signal form				
0	1	Duty-Cycle-Mode		Both channels (A / B) have the same frequency, but independently adjustable duty cycle				
1	0	Universal-Mode		Freely adjustable frequency & duty cycle for channel A, half frequency for channel B and 50% duty cycle				
1	1	Frequency-Mode		Freely adjustable frequencies for both channels (A / B), duty cycle is always 50%				

PWM-Period = Microcontroller-Base Clock / 2 / Prescaler / PWM0Ctrl1

PWM-Period = 16 MHz / 2 / 64 / 1000 = 125 Hz

De mode, het geactiveerde kanaal en de prescaler bits worden geselecteerd door een byte te creëren.

PS2	PS1	PS0	Description	Base Frequency
0	0	0	PWM-Ausgänge deaktiviert	-
0	0	1	Vorteiler (Prescaler): 1	16 MHz
0	1	0	Vorteiler (Prescaler): 8	2 MHz
0	1	1	Vorteiler (Prescaler): 64	250 kHz
1	0	0	Vorteiler (Prescaler): 256	62,5 kHz
1	0	1	Vorteiler (Prescaler): 1024	15,625 kHz
1	1	0	Vorteiler (Prescaler): 1024	15,625 kHz
1	1	1	Vorteiler (Prescaler): 1024	15,625 kHz

We hebben een frequentie nodig van 50Hz. Op basis van de formule wordt het volgende bekomen:

$$16\,000\,000 / 2 / 64 / 2500 = 50\text{Hz}$$

Dus nemen we als prescaler bits 011. De waarde van 2500 wordt doorgegeven als een 16bits woord bij de functie PWM0CTRL1.

We bekomen vervolgens voor PWM0CTRL0: 01101001

Voorlopig wordt enkel kanaal A geactiveerd om te testen.

1.1.2 PWM0CTRL1

Via deze functie wordt aangegeven door hoeveel de basisfrequentie moet worden gedeeld om 50Hz te bekomen. Hier wordt dit dus 2500 of 100111000100

1.1.3 PWM0CHA

De functiewaarde bepaalt de duty cycle. Voor de klauw bekomen we de volgende waarden:

MIN	NEUTRAAL	MAX
0,57ms	0,8ms	2,1ms
20ms / 0,57=35	20ms/0,8=25	20ms/2,1=9,52
2500/35=71	2500/25=100	2500/9,52=262
71	100	262

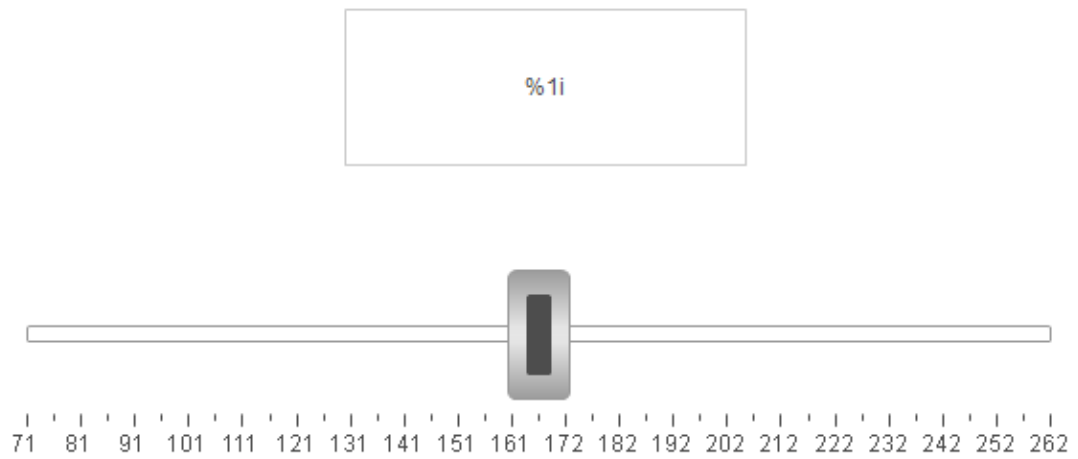
Deze waarden kunnen ook decimaal ingegeven worden:

Scope	Name	Address	Data type	Initialization	Comment	Attributes
 VAR_GLOBAL	GPIO24MASTER		BOOL			
 VAR_GLOBAL	PWM0CTRL1		WORD	2500		
 VAR_GLOBAL	AIO	%IW 100	WORD	71		
 VAR_GLOBAL	PWM0CHA		WORD	50		
 VAR_GLOBAL	PWMCTRL0		BYTE	105		

1.2 Webfunctionaliteit

Codesys biedt ook een webfunctionaliteit aan die op dezelfde manier functioneert als een HMI.

Deze kon ik 'lokaal' bereiken via het adres: <http://169.254.239.166:8080/webvisu.htm>



Het ontwerp ziet er sober uit, maar diende momenteel enkel als testomgeving.