Servo

7. januar 2023 21:20

Arduino kan sende pwm pulsbreddemodulation på flere frekvenser, som kan bruges til analogt output. Til servomotor bliver biblioteket servo.h brugt og det låser frekvensen på pwm til 50Hz. En servo bliver styret af pulsbredde, så det virker også logisk at bruge pulsbredde som driverens værdi. Der skal være en driver alene til servo formål.

Vær opmærksom på at Arduino bruger heltal til pulsbredde. Der er grænser for den præcision der kan ydes. Tilsvarende er de små standard servomotorer heller ikke særlig præcise. Er der behov for præcis postionering eller bevægelse, så se efter andre løsninger.

Arduino har 16bit opløsning på pin 9 og 10 og 8bit opløsning på de øvrige pins med PWM.

Bibliotek hedder: JBServoDrv.h.

Driver til servomotor

7. januar 2023 21:21

Driveren bliver sat op med motorens specifikationer. Driveren sender styresignal til en servomotor. Styresignalet kan være et interval, for bevægelse af motorens arm. Bevægelsen kan være ethvert vinkelinterval og følge ethvert bevægelsesmønster.

Funktion
Driver sættes op med en specifikation for servomotor og bevægelsesalgoritme.

Der er mulighed for at bygge et stykke specifikt software, der har sit eget mønster for bevægelse af servomotor. Den software beregner 1 specifik pulsbredde for hvert trin af bevægelsen og bruger driveren til at sende den pulsbredde ud på porten.

Driveren har sin egen algoritme for bevægelse. Ved denne brug af driver sættes i gang til en vinkeldrejning af servomotor, ved at indlæse parametre for bevægelsen, de overføres til beregningsalgoritmen, der initialiserer bevægelse. Fra næste klokkecyklus starter bevægelsen, som fortsætter indtil interval er gennemløbet. Når bevægelsen er udført hviler servomotor i slutposition, indtil en ny bevægelse bliver initialiseret.

Vær opmærksom på at en bevægelse ikke er helt præcis. Pulsbredde indstilles typisk i intervallet 530 - 2400µsek, hvilket bestemmer den højeste opløsning af bevægelsen. Det foretrækkes at bruge heltal i algoritme, for at spare program memory. Beslut om en bevægelse skal overholde et tidsinterval eller et vinkelinterval præcist. Forvent ikke at begge dele kan opnås.

Software begrænser bevægelse og pulsbredde, så servomotors specifikationer er overholdt.

For driverens design se softwarepattern facade.
For driverens design for tilkobling af beregningsalgoritme se softwarepattern strategy.

Fremtidige muligheder
Der kan være behov for at standse en bevægelse. For eksempel en motor til en skydedør. Motor bliver sat i bevægelse fra åben til lukket. Undervejs registrerer en sensor, noget er på vej gennem døren. Motor og dermed bevægelse bliver sat på pause, indtil døråbning er fri igen. I software er der flere løsninger. Klokkepulser til den timer der styrer bevægelsen kan blive stoppet. Alternativt kan der indbygges en pause og en continue metode i driver og beregningsalgoritme.

Opsætning
I bibliotek skal der erklæres globalt:
- unsigned int servoPeriod = REFRESH_INTERVAL/1000; // Se kildekode for servo.h.

Opsætningen fastlægger periodetid for pwm-port i msek.

I hovedprogram skal der erklæres globalt:
- Enum-liste med pin.
- t_ServoMotor <navn på servomotor-driver>

I hovedprogram setup() skal der opsættes:
- <navn på servomotor-driver>.begin(pin, motorSpecs)
- Opsætning af komponent til beregningsalgoritme.

Opsætning og brug i komponent
En komponent, der skal bruge en driver til en servomotor skal opsætte:
- Pointer til <navn på servomotor-driver>.

Klasse
Klassens navn: t_ServoMotor

Medlemmer
Servomotor-drivers medlemmer.

Navn	Type	Egenskab	Beskrivelse
servoPort	Servo		Kobling til pwm-port
motorSpecs	*struct ServoMotorSpecs		Pointer til servomotor specifikation
PWCalculator	*ServoMoveCalculator		Pointer til algoritme for bevægelse
isSetup	bool		Holder styr på om driver er initialiseret
fromPW	int		Bevægelse fra pulsbredde
toPW	int		Bevægelse til pulsbredde
currentPW	int		Nuværende pulsbredde opbevares her og leveres både til beregning og til port. Pulsbredde i µsek.
Seqs	enum	(STABLE, GOUP, GODOWN)	Sekvenser driver løber igennem
seq	Seqs		Nuværende sekvens

Metoder

Servomotor-drivers metoder.

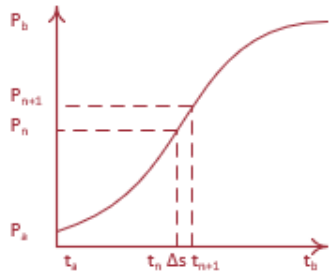
Navn	Argumentliste	Returnerer	Beskrivelse
begin	(pin: byte, motorSpecs: *ServoMotorSpecs PWCalculator: *ServoMoveCalculator)		Initialiserer driver til servomotor.
write	(pulseWidth: int)		Får opdateret driver med en specifik pulsbredde.
write	(fromAngle: int, toAngle: int, deltaTime: unsigned int, timeUnit: byte, sampleTime: unsigned int)		Modtager et vinkelinterval og gør klar til bevægelse af motorens arm. Tiden for bevægelse angives med deltaTime. Tidsenhed angives. Bevægelsens sampling periode angives med sampleTime. Default=servoPeriod. Tiden angives i msek.
doClockCycle	()		Ved behov får beregnet en ny pulsbredde og får opdateret pwm-port.
sendOut	(nextPW: int)		Opdaterer Arduino port. Porten bliver kun opdateret ved en aktuel ændring.

Algoritmer til bevægelse

22. april 2023 21:23

Der skal være mulighed for at servomotor kan tilpasses ethvert bevægelsesmønster. Servomotor softwarekomponent får tilknyttet en softwarealgoritmekomponent. Der er mulighed for i fremtiden at kode og tilkoble en ny softwarealgoritmekomponent.

Funktion
En bevægelse kan beskrives som følger:
Servomotor arm flytter sig fra en vinkel a til en anden vinkel b. Softwaren oversætter vinkel a til b til pulsbredde fra Pa til Pb. Der er en lineær konvertering fra vinkel til pulsbredde. Bevægelsen sker i tidsintervallet ta til tb. Bevægelsen opdeles i et antal samplinger med samplingstiden Δs.



Algoritme til bevægelse tænkes at følge formlen:

$$P(t_{n+1}) = P(t_n) + \Delta s * P'(t_n) + \frac{1}{2} * \Delta s^2 P''(t_n)$$

Pulsbredden svarer til en vinkel, der svarer til et sted, som begreb i fysikken for bevægelse.
En sampling ændrer fra forrige pulsbredde til næste pulsbredde, med hastighed udtrykt ved differentialet P'(t) og med acceleration udtrykt med dobbeltdifferentialet P''.
Overvejelserne her fører frem til de parametre, som algoritmen skal have overført, for at have tilstrækkeligt med oplysninger til enhver formel for beregning: Pa, Pb, tb-ta og Δs.
Algoritmen initialiseres hver gang en ny bevægelse skal udføres med de parametre, der er angivet. Derefter beregner algoritmen de koefficienter, hastigheder og accelerationer, der indgår i formlen og dermed er algoritmen klar til at levere pulsbredder. Algoritmen kan bruge en hvilken som helst formel, ikke kun den viste. Den angivne formel sporer tanken ind på de begreber der indgår i bevægelse.
Når bevægelsen er i gang kan en funktion levere $P(t_{n+1}) = f(P(t_n))$. $P(t_{n+1})$ opdateres i hvert samplingsinterval Δs. Funktionen skal kaldes i hver klokkecyklus.

Opsætning
Default er der indbygget en algoritme for lineær bevægelse. Hastighed er konstant i hele intervallet for pulsbredde. Algoritme hedder ServoLinearMove.

I hovedprogram skal der erklæres globalt:

- ServoMoveCalculator <navn på servomotor-kalkulator>

Klasse
Klassens navn: t_ServoMoveCalculator

Medlemmer
Grænseflade: Servomotor-kalkulator medlemmer.

Navn	Type	Egenskab	Beskrivelse
sampleTime	unsigned int		Samplingstid i beregning

Metoder
Grænseflade: Servomotor-kalkulator metoder.

Navn	Argumentliste	Returnerer	Beskrivelse
calCoefficient	(fromPW: int, toPW: int, deltaTime: unsigned int, sampleTime unsigned int)	int	Initialiserer beregninger. Modtager et pulsbreddeinterval. Tiden for bevægelse angives med deltaTime i msek. Bevægelsens sampling periode. Returnerer første pulsbredde. Pulsbredde i µsek.

calcNextPW	()	int	Udfører beregning og returner næste pulsbredde.

Default bevægelses-algoritme

30. januar 2023 11:22

En simpel algoritme for bevægelse er indbygget og bruges default.

Funktion
Bevægelsen er lineær og udføres med konstant hastighed.
Realtals beregninger er dyre i program memory, derfor bruger algoritmen heltals beregninger i stedet for. Et heltal ganges op med en præcision på 10, hvilket giver 1 decimals nøjagtighed. Det færdige resultat divideres med præcision. En heltals beregning runder en beregning ned.
En bevægelse bliver ikke helt præcis, men til gengæld bevares program memory, der er en knap ressource.

Algoritmen hedder: ServoLinearMove

Klasse
Klassens navn: t_ServoLinearMove

Medlemmer
Algoritmens medlemmer.

Navn	Type	Egenskab	Beskrivelse
PWSpeed	long		Pulsbredde ændring per sample. Pulsbredde i µsek.
currentPW	long		Aktuel pulsbredde i µsek.
precision	int	const	Sætter antal gange af heltal.
sampleTimer	t_SimpleTimer		Holder styr på tiden for næste pulsbredde

Metoder
Algoritmens metoder.

Navn	Argumentliste	Returnerer	Beskrivelse
calCoefficient	(fromPW: int, toPW: int, deltaTime: unsigned int, sampleTime unsigned int)	int	Initialiserer beregninger. Modtager et pulsbreddeinterval. Tiden for bevægelse angives med deltaTime i msek. Bevægelsens sampling periode. Returnerer første pulsbredde.
calcNextPW	()	int	Udfører beregning og returner næste pulsbredde.

Motorspecifikation

30. januar 2023 10:01

Hver type servomotor får sine specifikationer lagt ind i biblioteket.

Når en ukendt servomotor kommer til elektronikværkstedet, bliver den målt. Datablad og målinger giver som resultat servomotorens vinkelområde og pulsbreddeinterval. Derefter bliver motorens specifikationer lagt i biblioteket.

Opsætning i bibliotek
I bibliotek ligger en erklæring med struct for motorspecifikation. Se medlemmer herunder.

Struktur
Strukturens navn: t_ServoMotorSpecs

Medlemmer
En servomotors specifikationer ligger i en struct med følgende medlemmer.

Navn	Type	Egenskab	Beskrivelse
PulseWidthMin	int		Minimum pulsbredde
PulseWidthMax	int		Maksimum pulsbredde
AngleMin	int		Minimum vinkel
AngleMax	int		Maksimum vinkel