Tidsstyring, ure og timere

7. januar 2023 20:28

Polling bliver standard for tidsstyring, fordi det giver robust og overskuelig software.

Alle applikationer skal have dette kerne bibliotek, ellers er det en helt anden type applikation der er på vej.
Biblioteket indeholder udover ure og timere også diverse generelt anvendte funktioner.

I bibliotek skal der erklæres globalt:

- Enum-liste med slukket og tændt: OFF, ON.
- Enum-liste med tidsenhed: MSEC, SECONDS.
- Enum-liste med masterblinker: MASTERBLINKERNO=127.

Bibliotek hedder: JBKernel.h.

I de fleste Arduino applikationer er der kun behov for en tidsstyring med en simpel timer og en masterblinker. Det følger med kerne bibliotek.
Har en applikation behov for avancerede multivibratorer og timere, kan de kobles på. Tilkobling følger med bibliotek for multivibratorer og timere.

Tidsstyring

7. januar 2023 20:29

Arduino har en klokfrekvens på 16MHz og 32kb programmemory. Ved 32kb programmemory får hver adresse 16bit eller 2byte. Det vil sige med 16MHz kan arduino gennemløbe programmemory på 1msek. Dertil kommer adressering af datamemory. Den mindste periodetid er ca. 2msek.
Hele softwaremaskinen styres af polling med en periodetid på 5msek. Det er hurtigt nok til findeling af timeres periode. Alle timere er simple tællere.

Ur

7. januar 2023 20:30

Uret varetager klokkecyklus.
Klokkecyklus får en default værdi 5msek. Ved programmering kan specificeres en anden cyklus.

Funktionen pendulum holder Arduino på pause indtil en polling cyklus er nået. Derefter gennemfører Arduino en fuld rundtur i applikationen og vender tilbage til pendulum.

Navneområdets data
Uret har følgende statiske data.

Navn	Type	Egenskab	Beskrivelse
ClockCycle	byte	=5	Tiden for 1 klokkecyklus i msek

Funktioner
Urets metoder.

Navn	Argumentliste	Returnerer	Beskrivelse
pendulum	()		Der sørger for et præcist interval for hver polling cyklus
convertToClockCycles	(time: unsigned long)	unsigned long	Konverterer millisekunder til antal klokkecyklus

Omregning fra sekunder til millisekunder udføres med en makro.

```
#define SecondsToMilliSecs(A) A*1000
```

En simpel timer

7. januar 2023 20:31

Mange komponenter har brug for deres egen timer.
Denne timer er begrænset til tider op til 1 minut.

Funktion
Timer indstilles til en løbetid. Når tiden er udløbet, indstilles værdi til ON=1.
I næste klokkecyklus indstilles værdi igen til OFF=0.

En timer holdes kun synkron, når en metode kaldes i hver klokcyklus. Det er kun den komponent, der bruger en timer, der styrer den. Der er ingen central synkronisering af timere. Er der kun brug for en periode (oneshot), så stopper brugeren af timeren med at kalde dens metoder, når tiden er udløbet.

Klasse
Klassens navn: t_SimpleTimer

Medlemmer
Den simple timers medlemmer.

Navn	Type	Egenskab	Beskrivelse
noCycles	unsigned int		Antal klokkecyklus til udløb af tid
cycle	unsigned int		Nuværende klokkecyklus

Metoder
Den simple timers metoder.

Navn	Argumentliste	Returnerer	Beskrivelse
Constructor	(duration: unsigned int, timeUnit: byte)		Default bliver udløbstid sat beregnet ud fra tidsenhed
setDuration	(duration: unsigned int, timeUnit: byte)		Indstiller udløbstid sat beregnet ud fra tidsenhed. Nulstiller timer.
triggered	()	bool	Returnerer ON, når tiden er udløbet

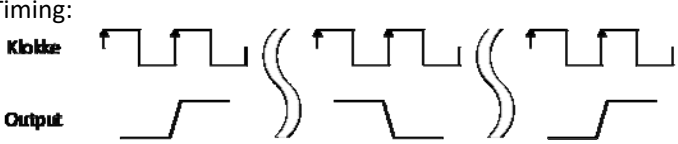
Master blinker

7. januar 2023 20:33

Til de fleste Arduino applikationer der skal bruge blink i lamper, er der en masterblinker.
Blinker kører konstant med fast periodetid.
Master blinker, ligger globalt, da der kun skal være 1.

Funktion
Blinker får en default værdi for halvperiode 500msek med dutycycle 50%. Ved programmering kan specificeres en anden halvperiodetid.

Timing:



Opsætning
Softwarekomponenter bruger en funktion, som tilkobling til masterblinker. Funktionen er indbygget i biblioteket, så den default compiles, når der ikke er defineret bibliotek med multivibratorer.

```
#ifndef Multivibrator_h
    blinkerNotification(blinkerNo=MASTERBLINKERNO).
#endif
```

Opsætning og brug i komponent
Komponenter der bruger masterblinker, skal kalde funktionen blinkerNotification(blinkerNo).

Navneområdets data
Masterblinker har følgende statiske data.

Navn	Type	Egenskab	Beskrivelse
HalfPeriod	unsigned int	=500	Varighed af en halvperiode i msek
value	bool		Masterblinkers værdi
timer	SimpleTimer	=HalfPeriod, MSEC	Timer der holder styr på tiden

Funktioner
Masterblinker er bygget ind i et "Namespace" med følgende funktioner.

Navn	Argumentliste	Returnerer	Beskrivelse
doClockCycle	()		Holder blinker kørende med hver klokkecyklus
dataOut	()	bool	Returnerer on/off med dutycycle 50%

Global funktion
Til globalt brug er der en funktion.

Navn	Argumentliste	Returnerer	Beskrivelse
blinkerNotification	(blinkerNo: unsigned int)	bool	Returnerer masterblinkers værdi. Nummeret skal pege på masterblinker. Er nummer ikke masterblinker bliver der returneret OFF.

Array funktioner

7. februar 2023 21:47

I biblioteker og i applikation er der samlinger af objekter, der ligger i arrays.
For at sikre imod at programmeringsfejl, får arduino til at fryse eller gå i loop, skal der indbygges tjek:

- Array index er gyldigt.
- At objektet på en plads i et array er initialiseret.

Enhver samling skal have et array "isSetup" af typen bool, som er true, hvis et objekt er initialiseret.

Funktioner
Støttefunktioner ligger globalt.

Navn	Argumentliste	Returnerer	Beskrivelse
isValidIndex	(index: unsigned int, arrayLength: unsigned int)	bool	Returnerer om et arrayindex er gyldigt
hasConfig	(isSetup[]: bool, unsigned index: int, arrayLength: unsigned int)	bool	Returnerer om et objekt er initialiseret. Setup er lagt i et separat array.
hasConfig	(*element: void, index: unsigned int, arrayLength: unsigned int)	bool	Returnerer om et objekt er initialiseret i et array med pointere.