

Säkerhet och automation - Larmsystem med MD407 och en CAN buss

Kevin Cheng, Albin Essman
Tony Le, Caesar Paulsson
Daniel Melin Romstad, Hugo Söderbergh

30 augusti 2026

Innehåll

1	Inledning	1
1.1	Syfte	1
1.2	Mål	1
1.3	Arbetsmetod	2
1.3.1	Teknisk utveckling	2
1.3.2	Verifikation	2
1.3.3	Filhantering och dokumentation	2
2	Teknisk beskrivning	3
2.1	Teknisk bakgrund	3
2.2	Översikt av MD407-kortet	3
2.2.1	GPIO	3
2.2.2	USART	3
2.2.3	CAN-buss	4
2.2.4	CAN protokoll	4
2.2.5	Timer	4
2.2.6	Avbrott	5
2.3	Komponentöversikt	6
2.3.1	Knappsats	6
2.3.2	Vibrationssensor	7
2.3.3	Avståndsmätare	7
2.3.4	Applikationsprotokoll	7
2.4	Systemöversikt	8
2.4.1	Centralenhet	8
2.4.2	Dörrlarmsenhet	8
2.4.3	Rörelselarmsenhet	9
2.4.4	Störenhet	10
3	Resultat	10
4	Diskussion	11
4.1	Syftet och målet	11
4.2	Metodval	11
4.3	Möjliga tekniska förbättringar	11
4.4	Möjliga förbättringar i utförande	12
	Referenser	13
	Bilaga A Kravspecifikation	
	Bilaga B Uppföljning av kravspecifikation	
	Bilaga C Tabell av tester och resultat	
	Bilaga D Test av knappsats modulen	
	Bilaga E Test av read_code modulen	
	Bilaga F Test av kalibration hos vibrationssensor	

Bilaga G Test av kommunikation mellan två MD407 kort via CAN-buss

Bilaga H Test av ping meddelande från vardera periferienhet till centralenhet

Bilaga I Test av UltraljudSensor - precision av uppmätning

Bilaga J Test för initiering av Rörelseenhet via CAN

Bilaga K Test av Störenheten

Bilaga L Test av sensorsystemets usart-state

Bilaga M Test av prototyp

Bilaga N Kommunikation och Samordningp

Tekniska termer

ARM (Advanced RISC Machine) är en arkitektur för processorer.

Avbrott är då exekveringen av programmet hoppar någon annanstans tillfälligt, vilket kan triggas av t.ex. hårdvarutimers.

CAN (Controller Area Network) är en buss och ett protokoll för dataöverföring.

EXTI (External interrupt/event controller) är det som kontrollerar avbrott från GPIO i MD407.

FIFO (First In First Out) är ett koncept som lyder att datan som kommer in först är det som har högst prioritet.

GPIO (General Purpose Input-Output) är ett sätt att kommunicera mellan MD407 och annan hårdvara.

MD407 är en mikrodator baserad på ARM arkitekturen.

NVIC (Nested Vector Interrupt Controller) är det som styr avbrotten i MD407.

RJ11 (Registered Jack 11) är en kabel för att överföra data mellan enheter.

USART (Universal synchronous and asynchronous receiver-transmitter) är ett seriellt gränssnitt för att kommunicera mellan enheter.

USB (Universal Serial Bus) är ett seriellt dataöverföringsprotokoll och kontakt.

1 Inledning

Inbrott är en händelse som kostar privatpersoner obehag och ekonomiska kostnader. År 2021 rapporterade 1.3% av alla hushåll i Sverige att de hade utsatts för bostadsinbrott, vilket motsvarar 60 000 hushåll [1]. Det visade sig att det genomsnittliga inbrottet kostade 22 000 kr i både skador och stulen egendom för försäkringsbolaget If:s kunder 2020 [2]. Om vi förutsätter att detta är en representativ siffra för kostnaden av inbrott generellt innebär detta att inbrott kostade 1.3 miljarder kronor år 2021. Det är dock inte bara en ekonomisk kostnad, utan även en emotionell kostnad, då folk som utsatts för inbrott senare är rädda för fler inbrott och ändrar sina vanor som respons [3].

Det är därför viktigt att motverka effekten av inbrott. Att förebygga inbrott helt och hållet skulle vara det optimala, men detta är inte möjligt i praktiken. Detta innebär att en strategi av att försöka förebygga så många inbrott som möjligt, samt minska kostnaderna för när inbrott väl händer bör tillämpas. Det finns många åtgärder som man kan göra som privatperson för att minska risken för inbrott, så som att ha utomhusbelysning med rörelsesensor eller att ha någon som tar hand om ens post då man reser bort [4].

En annan åtgärd som minskar kostnaderna för inbrott är att installera ett larmsystem. Det är inte klarlagt om ett hemlarm har en förebyggande effekt, men det har en skademildrande effekt [5]. Detta är eftersom ett larm kan avbryta ett pågående inbrott och på så sätt minimera skadorna. Detta innebär att det finns ett behov av larmsystem som kan detektera inbrott och till följd starta ett larm som sedan kan kopplas vidare till andra åtgärder.

1.1 Syfte

Projektet syftar till att bygga ett larmsystem som ska kunna detektera inbrott. Systemet skall bestå av ett antal mikrodatorer som kommer vara utrustade med dörrsensorer, avståndsmätare, och vibrationssensor. Detta ger en insikt i hur ett larmsystem är uppbyggt och vilka möjligheter och begränsningar som finns, samt vilka säkerhetsåtgärder som måste tas för att se till att systemet inte enkelt kan kringgås av en inkräktare.

1.2 Mål

Projektet innebär konstruktion av ett larmsystem. Systemet är uppbyggt av mikrodatorer uppdelade i en hierarki med flera periferienheter anslutna till en centralenhet, som ansluts till extern dator. Periferienheterna ska hantera sensorer och lokalt larm, samt kommunicera med centralenheten som ansvarar för kommunikation med externa enheter. Utöver enheters individuella specifikationer ska systemet uppfylla grundläggande krav kring säkerhet och pålitlighet.

Övriga mål som systemet ska försöka nå är förmågan att kunna testa periferienheter med en anslutning till extern dator som simulerar centralenheten. Periferienheten ska även kunna larmas av med en knappsats ansluten till periferienheterna. Utöver dessa funktioner kan ett system för ljuduppspelning vid larm utvecklas, om tid för utveckling återstår.

1.3 Arbetsmetod

I den här delen av rapporten redogörs för de arbetsmetoder som har utnyttjats för att uppnå projektets olika mål och beskriver även hur utvecklingen har gått tillväga. För hur kommunikation och samordningen av projektet gick tillväga, se bilaga N.

1.3.1 Teknisk utveckling

Den tekniska utvecklingen av projektet var primärt uppdelad i olika moduler för att lämpligt fördela gruppens arbetskraft och få ut så mycket effektiv arbetstid som möjligt. Dessa olika moduler bestod inte bara av mjukvara som t.ex. CAN-bussen (2.2.3) men även periferienheter som exempelvis vibrations-sensorn (2.3.2) och knappsatsen (2.3.1) som krävde sina enskilda drivrutiner och funktioner. Dessa individuella moduler var givna till enskilda gruppmedlemmar och denna arbetsuppdelning visade sig vara effektiv, eftersom de gruppmedlemmar med mer erfarenhet inom kodutveckling arbetade på de lite mer komplexa moduler och vice versa. Efter dessa var klara kopplades de ihop till ett helt larmsystem. Dessutom kodades projektet helt i programmeringsspråket C, då det var mest effektivt att använda med MD407-enheter.

1.3.2 Verifikation

Tester och undersökningar på systemet fördes under projektets utveckling för att se till att systemet uppfyllde de kraven som ställdes. Testerna följde ett systematiskt och tydligt utförande för att tillgodose konstruktionen av precisa och välgjorda tester.

Dessa tester utnyttjades under projektets utveckling främst på systemets olika moduler, vilket var bestående av både hårdvara och mjukvara. Se Bilagorna för en mer detaljerad beskrivning av de metoder som användes, resultat och analys som framställdes i dessa tester.

1.3.3 Filhantering och dokumentation

Filhantering och synkronisering mellan medlemmar sköttes helt med hjälp av filhanteringsprogrammet Git. Git valdes på grund av dess enkla sätt att ladda upp och ner nya versioner av filer som gjorts. Git innehar även sätt att dela upp filerna på i olika "branches", så att det inte uppstår problem när en utvecklare skriver i en fil som använder kod som fortfarande uppdateras av en annan utvecklare. Detta bidrog till en smidigare samarbetsprocess samt ett effektivt projektgenomförande.

2 Teknisk beskrivning

Projektets tekniska delar kommer att introduceras med en bakgrund som ger en överblick av det som fanns till användning under systemets utveckling. Därefter beskrivs de tillgängliga komponenterna samt de olika mjukvarukoncept som har skapats för systemet.

2.1 Teknisk bakgrund

Till projektets utveckling har tre ARM-baserade MD407 Rev4 mikrodatorer som gjorts tillgängliga för användning. Dessutom var även andra tillbehör tillgängliga bestående av en vibrationssensor, en ultraljudsbaserad avståndsmätare, en knappsats, en sjusegmentsdisplay, en mängd ljusdioder, en samling av olika kablar samt en kopplingsplatta. MD407-kortet förfogas för projektet på grund av utvecklarnas tidigare erfarenheter med mikrodatorn.

Ytterligare är larmsystemet programmerat i programmeringsspråket C och med hjälp av funktioner ur det standardbiblioteket STM32F4xx har kod skrivits till MD407-kortet för att driva bland annat CAN-bussen och de olika portarna i mikrodatorn.

2.2 Översikt av MD407-kortet

MD407 är en enkorts dator baserad på ARM med en Coretex M4 ARMv7 processor. Mikrodatorn har stöd för kommunikation över CAN, Ethernet och USB. CAN kommunikationen sker i MD407-kortets två CAN-portar med RJ11-kontakt och för USB finns det två USB-portar av typ A och B. Utöver detta finns även fem GPIO-portar genom pin-arrays för kopplingen av de olika komponenter som finns tillgängliga.

2.2.1 GPIO

Kortet är försett med ett antal uppsättningar pinnar som kan användas för att kommunicera med kortet från annan hårdvara. Dessa uppsättningar pinnar kallas GPIO portar (General Purpose Input-Output), och det finns fem stycken, namngivna GPIO A till GPIO E. Varje GPIO port har 16 stycken pinnar, som kan ställas in som in-pinnar eller ut-pinnar. Genom att ställa en pinne som en ut-pinne går det att mata ut ström över pinnen, och om den ställs som in-pinne går det att läsa av spänningen som går in i pinnen. Om en pinne ska läsas av är det ofta eftersträfvansvärt att den läsas som låg eller hög om spänningen är flytande. För detta ändamål går det att ställa in ett s.k. pull-up eller pull-down motstånd på pinnen, vilket gör just detta.

GPIO portarna används i samtliga enheter i projektet, för olika ändamål. I centralenheten (2.4.1) används GPIO för att kommunicera med knappsatsen. Rörelseenheten (2.4.3) använder GPIO för att kommunicera med vibrationssensor såväl som avståndsmätare och dörrenheten (2.4.2) använder GPIO för att läsa av dörrsensorernas värde, samt för att ge status genom lysdioder.

2.2.2 USART

USART, vilket står för Universal Synchronous and Asynchronous Receiver-Transmitter används för att skicka ut data seriellt från centralenheten till en

persondators terminal och även för att skicka över data från datorn till centralenheten som läser av och sparar det. Detta är hur konfigureringen av larmer görs via en dator. Under utvecklingen var USARTs enkla system för att ladda över program viktig då kod enkelt kunde testas.

2.2.3 CAN-buss

CAN (Controller Area Network) är en kommunikations bus mellan enheter som skickar data. CAN använder sig av olika protokoll där den undviker att data försvinner eller blir korrupt. Dessa protokoll är uppdelade i block där användaren kan avkoda felmeddelanden och säkerhetsställa inkommande meddelanden. Nätverket som CAN bussen utgör är halv duplex som innebär att bussen endast tillåter en nod att skicka data åt gången som därmed alla enheter i nätverket mottas.

2.2.4 CAN protokoll

I djupare detalj består CAN protokoll av två ledare som motsvarar värdet av bussen. Ledarna - CAN LOW och CAN HIGH, är kopplade tillsammans med alla enheter i kretsen och det som avgör vad som skickas är spänningsskillnaden mellan vardera ledare. Vid ingen spänningsskillnad tillförs en logisk etta och vid spänningsskillnad högre än 2 skickas en logisk nolla. Vilket ger bussen endast två värden.

I CAN protokoll finns det regler för vilken data som skickas först. Detta innebär att bussen inte följer FIFO (First In First Out) konceptet utan prioriterar vissa meddelanden före andra. Det är meddelandets värde som avgör dess prioritet. I CAN har logisk 0:a högre prioritet än logisk 1:a som betyder att ett mindre värde på meddelandet ger högre prioritet. Det här används för att undvika kollisioner av meddelanden på bussen men samtidigt skapa en hierarki på vilka meddelanden som skall prioriteras före andra.

I protokollet kan det framföras som väldigt tillförlitlig då den innehåller säkerhets mekanismer i dess CAN-frame. Hela ramen kan huvudsakligen delas in i fem olika säkerhetssystem: bit stuffing, bit monitoring, frame check, acknowledgement check, and cyclic redundancy check (CRC).

Vid tillkoppling av säkerhetssystemet används två CAN bussar av hårdvaran RJ11-kontakt som kopplas mellan centralenhet och båda periferienheter.

2.2.5 Timer

Vid hantering av tidsrelaterade avbrott och händelser har MD407-kort inbyggda timerdrivrutiner. En timerdrivrutin kan kontinuerligt utlösa funktionaliteter i olika tidsintervaller. I larmsystemet implementeras alla enheter med timers. De drivrutiner som använder sig av tidsbaserade händelser är sändning och mottagning av ping, avståndsmätaren vid mätning av puls och kontroll av öppna dörrar.

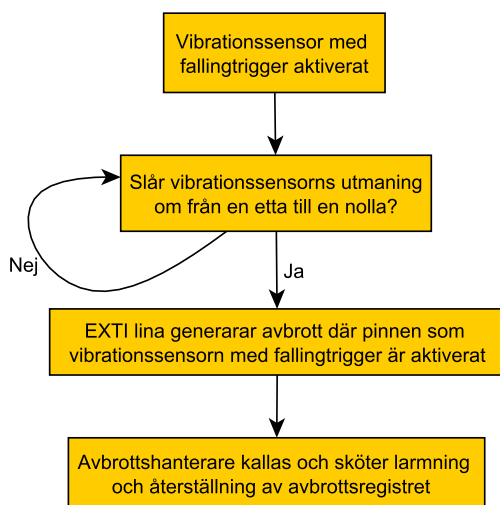
2.2.6 Avbrott

MD407-kortet har stöd för flera olika hårdvaruavbrott vilket användes för till exempel vibrationssensorn (2.3.2) och avståndsmätaren (2.3.3). Dessa olika avbrott har en stor medverkan på tidsbaserad och händelsestyrd bearbetning.

Ett av de kännetecknande avbrotten i MD407-kortet är NVIC vilket står för Nested Vector Interrupt Controller. NVIC hanterar avbrottets prioritet vilket förbättrar MD407-kortets prestation och förminskar avbrottets åtkomsttid. Det här görs genom NVICs tilldelning av en prioritetsnivå till varje avbrottskälla. Om ett avbrott med högre prioritetsnivå aktiveras då ett avbrott med lägre prioritetsnivå är i tjänst, så ger NVIC det hög prioritetsavbrottet friheten att aktivera och ta förtur.

Ytterligare ett kännetecknande avbrott i MD407 mikrodatorn är EXTI som står för External Interrupt. EXTI skapar möjligheten att generera avbrott som är baserade på externa händelser genom GPIO pinnar. Detta görs genom att konfigurera EXTI linor genom att specificera vilken GPIO pin/pinnar som är associerad med vardera EXTI lina. Triggvillkorn (vilka händelser) för EXTI linan specificeras också vilket tillåter avbrott baserad på externa händelser.

EXTI avbrott tillåter bland annat att vibrationsensorn implementeras utan användning av blockerande funktioner. Hur denna processen gick tillväga demonstreras i figur 1:



Figur 1: EXTI avbrott genereringen för vibrationssensorn

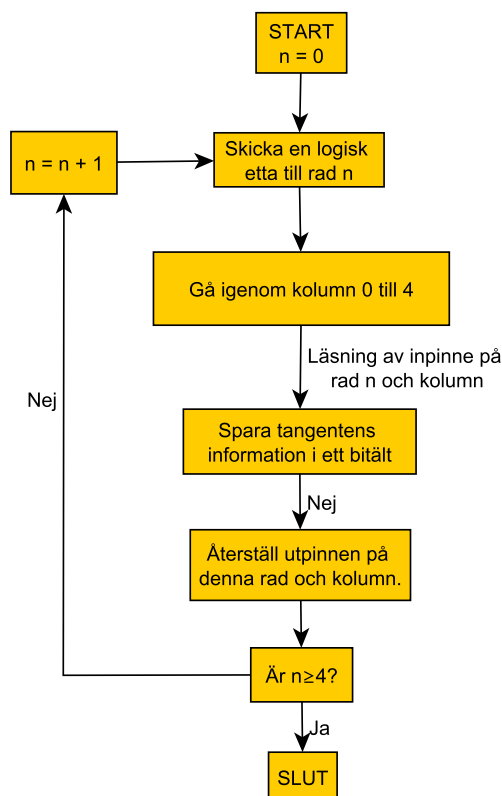
På detta sätt kan det alltid upptäckas när vibrationer detekteras utan att andra funktioner av larmsystemet komprimeras.

2.3 Komponentöversikt

Utöver den tekniska bakgrunden beskrivs även de olika komponenterna mer detaljerat nedan. Dessutom tydliggörs komponenternas funktion i projektet och hur de implementeras i koden.

2.3.1 Knappsats

Knappsatsen kommer att användas för att stänga av larmet från centralenheten då det gått igång. Det ska då gå att knappa in en kod som bestäms under systemkonfiguration, varvid larmet skall avaktiveras. Knappsatsen har 4 inpinnar för raderna och 4 utpinnar för kolumnerna. För ett överblick av hur knappsatsen tangenterna lagras, se figur 2:



Figur 2: Lagring av knappsatsen

För att läsa om knappen på rad n och kolumn k är nedtryckt, skickas först en utsignal till n :te utpinnen, och därefter läses den k :te inpinnen. Om värdet då är högt är knappen nedtryckt annars är den inte det.

För att se vilka tangenter som nyligen trycktes ned jämförs detta resultat med vad vi läste förra gången. Om en tangent lästes som icke-nedtryckt förra gången men är nedtryckt denna gången, tolkas detta som att tangenten just trycktes ned, vilket sparas i ett till bitfält för nyligen nedtryckta tangenter.

2.3.2 Vibrationssensor

Vibrationssensorn har två pinnar avsedda för ström och jord samt två utpinnar, en analog och en digital. Den analoga utpinnen ger ett flytande värde som påverkas av styrkan på vibrationen medan den digitala utpinnen visar en nolla under detekterad vibration och en etta när ingen vibration detekteras. Den digitala utpinnen kommer användas då dennes utmatning är enklare att implementera. Sensorns känslighet går att justera manuellt direkt på enheten till den nivå som behövs.

2.3.3 Avståndsmätare

Avståndsmätaren av modell HC-SR04 består av en ultraljudssändare, en mottagare samt en kontrollkrets. Sändaren skickar ut åtta pulser med en frekvens på 40kHz, som då reflekteras tillbaka till mottagaren efter kollision med objekt. Mätaren har en räckvidd från 2cm till 400 cm och en exakthet upp till 3mm.

Mätaren har totalt 4 pinnar för anslutning till enheter, två av dessa för strömöverföring i form av 5v ingång samt jord, och två för aktivering och läsning av mätningar i form av en 'trigger' inpinne, samt en 'echo' utpinne. För att aktivera en mätning skickas en kort puls på 10 μ s, vilket aktiverar sändaren. När signalen har träffat mottagaren aktiveras mätarens utpinne, som är aktiv lika länge som det tog för signalen att färdas till sändaren. Då ljudets hastighet i luft är ungefär 340m/s kan vi omvandla tiden till avstånd.

2.3.4 Applikationsprotokoll

Meddelanden som skickas via CAN kommunikation använder sig av definierade protokoll och strukturer. Varje CAN meddelande följer en standardiserad datastruktur men skilljer sig i själva innehållet. Figur 3 visar en förenklad representation av ett CAN-meddelandes struktur:

Message ID	Data Length	Data (0-8 bytes)
------------	-------------	------------------

Figur 3: Meddelande Struktur

Meddelande-ID representerar identifikationsnumret för noden som överför meddelandet. Data-längden anger antalet karaktärer i data-listan och data-listan som är själva data innehållet i meddelandet. Den initiala karaktären i data-listan fungerar som ett unikt funktions-ID. Varje funktion tilldelas ett distinkt ID-nummer där exempelvis ID 0 är reserverat för alarm och ID 1 används för ping. De efterföljande karaktärerna i listan beskriver de parametrar som är associerade med den specifika funktionen.

2.4 Systemöversikt

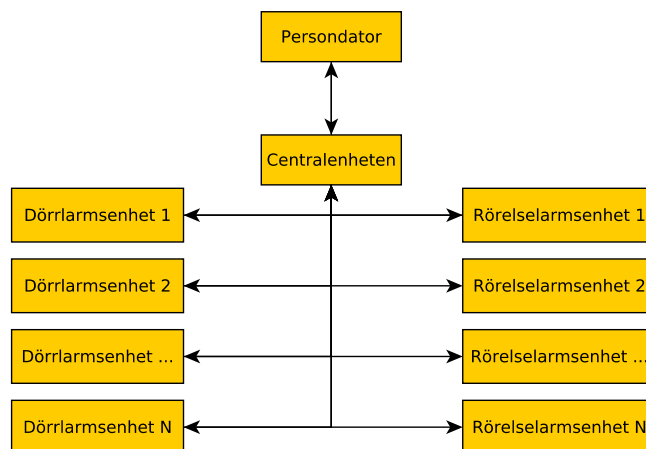
Systemet består av ett antal MD407 mikrodatorer med olika roller. Det finns en centralenhet som kopplas till ett flertal periferienheter i form av dörrlarmsenheter och rörelselarmsenheter (se figur 4). Dessa enheter har i uppgift att detektera intrång med hjälp av sensorer som kopplats till dem (se figur 5 och 6). För att kommunicera med centralenheten använder sig enheterna sig av en CAN buss.

2.4.1 Centralenhet

Centralenheten hanterar alla dörrlarmsenheter och rörelselarmsenheter, samt kontrollerar när larmet är utlöst. För att kunna stänga av larmet finns det en knappsats kopplad till centralenheten där användare kan identifiera sig med en kod för att deaktivera larmet. Larmet aktiveras om:

- en av dörrlarms eller rörelselarmsenheter triggat larmet
- en av enheterna kopplas bort från nätverket

Centralenheten kopplas även via USART till en persondator för att kunna konfigurera systemet via datorn. Denna kommer även att spela upp ett obehagligt larm ljud då larmet har gått. För en översikt av centralenheten se figur 4:

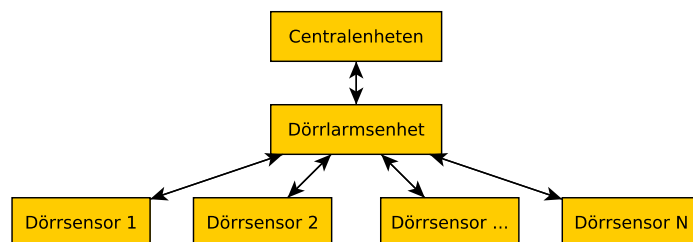


Figur 4: Centralenheten kopplas till en persondator och ett antal periferienheter

2.4.2 Dörrlarmsenhet

Dörrlarmsenheterna har ett antal dörrsensorer som känner av om en eller flera dörrar är öppna. Om minst en dörr lämnas öppen i fyra minuter larmar dörrlarmsenheten lokalt tills de larmande dörrarna stängts, vilket innebär att en varningslampa tänds. Om en av dörrarna lämnas öppen ytterligare en minut larmar dörrlarmsenheten till centralenheten, vilket orsakar att huvudlarmet går. Om en dörrsensor kopplas från dörrheten kommer detta märkas som att dörren lämnats öppen till följd av att de är kopplade med pull-down motstånd.

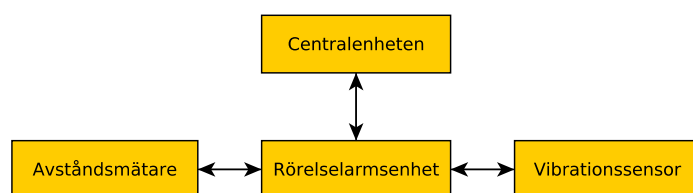
Det kommer att finnas maximalt 15 stycken dörrsensorer eftersom det är det är så många dörrsensorer som kan stödjas maximalt med hårdvaran då det även behöver finnas en lampa för varje dörr, samt en för larmdioden. Varje dörrsensor ges ett unikt id för hantering av centralenheten. Antalet dörrar konfigureras då dörrlarmsenheten startas via centralenheten. För en översikt av dörrlarmsenheten se figur 5:



Figur 5: En dörrlarmsenhet övervakar ett antal dörrsensorer

2.4.3 Rörelselarmsenhet

Rörelselarmsenheterna är lika dörrlarmsenheterna i sitt beteende. De har endast en avståndsmätare och en vibrationssensor per enhet, då detta bedömdes vara tillräckligt för att de skulle uppfylla sitt syfte. Detta gjorde även programmeringen enklare då endast läsningen av avståndsmätaren är tidskänlig eftersom inläsningen av vibrationssensorn sker via avbrott. Rörelselarmsenheten larmar till centralenheten om något kommer för nära avståndsmätaren, där avståndet bestäms när enheten startar, men går att omkalibrera med en funktion som kallas via USART från centralenheten. Den kommer även att larma om vibrationssensorn känner av vibrationer enligt känsligheten som kan ställas in på den fysiska komponenten. För en översikt av dörrlarmsenheten se figur 6:



Figur 6: En rörelselarmsenhet kopplas till en avståndsmätare och vibrationssensor

2.4.4 Störenhet

För att kunna testa tillförlitligheten i CAN kommunikationen skapades även en störenhet. Störenheten simulerar en defekt periferienhet genom att skicka ut stora mängder data över CAN nätverket. Mängden data kan kontrolleras för att testa nätverket vid olika belastningsgrader. Detta tillåter oss att se till att kritiska meddelanden så som larm-meddelanden fortfarande når fram över CAN bussen när nätverket är tungt belastat.

3 Resultat

Projektet resulterade i ett fungerande larmsystem med rörelselarmsenhet, dörr-larmsenhet, störenhet samt centralenhet. En kravspecifikation gjordes för att kunna kontrollera att larmet uppfyllt de önskade funktionerna. Denna finnes i bilaga A och en uppföljning för kravspecifikationen finnes i bilaga B. Ytterligare demonstrerar uppföljningen av kravspecifikationen att larmet uppfyllde alla önskade funktioner. Utöver detta utfördes även tester på de olika modulerna för att säkerställa att de fungerar som de ska, dessa finnes numrerade som bilagor D till M. Dessutom finns det en tabell under bilaga C som sammanställer alla tester från bilagor D till M och ger deras resultat samt analys. Denna tabellen visar att alla tester som genomfördes var en succé och framställde att de olika modulerna som följde de förväntningarna som gruppen hade.

Rörelselarmsenheten är kopplad till en vibrationssensor och en rörelsesensor. Dessa har samma uppgift, att känna av om något rör sig i mätområdet. Rörelsesensorn gör det via ljud medan vibrationssensorn känner av vibrationer. Dörrlarmsenheten är kopplad till dörrsensorer som känner av om en dörr öppnas samt hur länge den lämnas öppen. Efter att en tidströskel nås larmar enheten. Störenhetens uppgift är att testa tillförlitligheten på systemet genom att simulera en defekt periferienhet som skickar stora mängder data över CAN bussen. Centralenheten konfigurerar larmet och kommunicerar med hjälp av CAN periferienheterna. Centralenheten är även kopplad till en knappsats som används för att avaktivera larmet.

4 Diskussion

I detta kapitel diskuteras projektet om dess syfte, mål, utförande samt möjliga förbättringar.

4.1 Syftet och målet

Syftet med detta projekt var att skapa ett larmsystem som detekterar larm som kan triggas med inbrott via dörröppning, vibration samt rörelse. Resultatet av projektets slutförande är ett välfungerande larmsystem som har de tidigare nämnda sakerna som fungerar och som kan samverka i larmsystemet. Samtidigt som själva projektet formaterades gjordes extra funktioner till larmsystemet. De extra funktionerna gäller då att man kan ändra mellan USART och CAN läge för felsökning. Detta har en stor fördel för att identifiera fel i implementation i programmet och underlättar mycket vid testkörning. Ytterligare har vi konstruerat ett larm med ljudspelning som körs vid aktiverat larm.

4.2 Metodval

Detta projekt tog längre tid än förväntat då många svårigheter framkom under projektets utförande. Det skedde bland annat gällande hur kommunikationsprotokollet skulle struktureras och hur dörrenheten skulle fungera. Implementation av kommunikation tog längre tid än förutsatt då en gemensam struktur för meddelandet behövdes samt att vi inte visste hur systemet skulle se ut. Det uppstod även problem vid sammansättningen av modulerna med diverse buggar samt ej kompatibel kod, vilket hindrade projektets utveckling.

Arbetsordningen som gick ut på att separat utveckla de olika komponenterna för att sedan sätta ihop dem var effektiv på det sätt att den utspridda utvecklingen möjliggjorde tidig identifiering av de problem som fanns hos komponenterna. Den utspridda utvecklingen var dock inte särskilt tidseffektiv vilket ledde till en sen ihopkoppling och då mindre tid till utveckling av hela systemet.

4.3 Möjliga tekniska förbättringar

Trots att alla tekniska mål blivit uppfyllda finns det fortfarande rum för förbättring av systemet, framförallt kring CAN kommunikationen. Den första förbättringen som skulle förbättra kommunikationen är att hantera sändningsfel, då vi i nuvarande systemet inte tar hänsyn till att CAN kanalen skulle kunna vara opålitlig. Det finns också många förbättringar kring säkerhet av kontrollmeddelanden som kan göras. En av dessa skulle vara att dela upp sändar id så att larmmeddelanden går över ett id med högre prioritet, och därför inte störs ut lika lätt. Ett annat säkerhetsproblem är att en inkräktare trivialt kan avaktivera alla enheter i systemet om de får tillgång till sändning av meddelanden över CAN, vilket skulle kunna lösas genom att verifiera att det är centralenheten som skickat meddelandet genom kryptografi. Det finns även fler säkerhetsaspekter som kan tillämpas såsom skydd mot replay-attacker och verifiering av ping-meddelanden.

4.4 Möjliga förbättringar i utförande

Projektutförandet skedde inte som planerat och många ändringar och idéer förekom under genomförandet. Mycket av det skulle kunna underlättats genom en bättre planering för projektet samt ett bättre samarbete och tilldelning i projektarbetet.

En till möjlig förbättring skulle vara att inte fördela moduler en och en, men istället parvis. Detta eftersom det ger möjligheten att bygga på varandras idéer och därmed lättare ta projektet framåt. Även eftersom det hände att moduler behövde att andra gjordes först hade det varit fördelaktigt om de kunde arbetats på parvis.

Något mer som kunde gjorts bättre var att sätta ihop delar tidigare då det blev lite stressigt i slutet, anledningen till det var att det antogs vara lättare att synkronisera delarna än det faktiskt var.

Referenser

- [1] BRÅ (Brottsförebyggande rådet), “Nationella trygghetsundersökningen,” <https://bra.se/publikationer/arkiv/publikationer/2022-10-11-nationella-trygghetsundersokningen-2022.html>, 2022.
- [2] If, “Bostadsinbrott,” Försäkrings bolaget If. (2023). Bostadsinbrott. Hämtad från <https://www.if.se/om-if/if-i-samhallet/vardagstrygghet/inbrott>, hämtades 4 september 2023.
- [3] E. Wessel and L. Nordlund, “Det har blivit ett fängelse”: En kvalitativ studie om hur vardagslivet för äldre personer påverkas till följd av direkt och indirekt viktimisering av inbrott,” Ph.D. dissertation, Mittuniversitetet, 2017, 2017-06-01. [Online]. Available: <https://urn.kb.se/resolve?urn=urn:nbn:se:miun:diva-31982>
- [4] polisen.se, “Bostadsinbrott - skydda dig,” <https://polisen.se/utsatt-for-brott/skydda-dig-mot-brott/stold-och-inbrott/bostadsinbrott/>, hämtades 18 september 2023.
- [5] Larm.se, “Har hemlarm någon effekt?” <https://larm.se/effekten-av-ett-inbrottslarm>, hämtades 4 september 2023.

A Kravspecifikation

Tabell 1: Kravspecifikation

Enhet	Funktion	Beskrining
Rörelserenhet	PING	Skicka ping till centralenheten.
Rörelserenhet	USART Debug	Kunna debugga via usart.
Rörelserenhet	USART Setup	Setup USART.
Rörelserenhet	Skicka alarm	Skickar alarm om någon sensor triggas.
Rörelserenhet	Ta emot alarm	Ta emot larm status.
Rörelserenhet	Mätavstånd	Mät avstånd och larma ifall något detekteras.
Rörelserenhet	Vibrationssensor	Larma om vibration detekteras i närhet.
Rörelserenhet	Sensor Init	Initierar rörelseenheten.
Rörelserenhet	Set Variation	Ger variation på avståndsmätare.
Rörelserenhet	Akivering via CAN	Aktiverar larmhantering.
Rörelserenhet	Sensor status	aktivering och deaktivering av sensorer.
Rörelserenhet	Can_ID	Ger id till rörelseenhet.
Rörelserenhet	Kalibrering	Kalibrera avståndsmätare via CAN.
Dörrenhet	PING	Skicka ping till centralenhet.
Dörrenhet	USART Debug	Debugga dörrenhetens funktionaliteter via usart.
Dörrenhet	USART setup	Setup USART.
Dörrenhet	Skicka Alarm	Skicka alarm om dörr varit öppen under lång period.
Dörrenhet	Ta emot alarm	Ta emot larm status.
Dörrenhet	Door Setup	Initialiserar dörrenheten vad antal dörrar, lokala timer och globala timer.
Dörrenhet	SET_DOOR_STATUS	Ändra specifik dörrstatus om den ska vara larmad eller inte.
Dörrenhet	Kontrollera dörrstatus	Ge lokalt alarm då minst en dörr varit öppen under kort tid och skickar globalt alarm efter en längre tid.
Dörrenhet	Can_ID	Ger id till dörrenhet.
Centralenhet	USART Setup	Setup USART.
Centralenhet	USART funktionshantering	Usart funktioner för konfiguration av periferienheter.
Centralenhet	Central alarm	Ljud uppspelning via usart och python. Globalt alarm aktiv.
Centralenhet	Keypad	Keypad för avlarmning.
Centralenhet	Ta emot ping	Larma om enhet inte är ansluten.
Centralenhet	Ta emot larmstatus	Ta emot larmstatus.
Centralenhet	Lagra information om periferienheter	Sparar information om periferienheterna i en lista.
Störenhet	USART setup	Setup USART.
Störenhet	Test av störenhet	Skickar slumpmässig data via CAN bus.

B Uppföljning av kravspecifikation

Tabell 1: Uppföljning av kravspecifikation

Enhet	Funktion	Beskrining	Status
Rörelserenhet	PING	Skicka ping till centralenheten.	X
Rörelserenhet	USART Debug	Kunna debugga via usart.	X
Rörelserenhet	USART Setup	Setup USART.	X
Rörelserenhet	Skicka alarm	Skickar alarm om någon sensor triggas.	X
Rörelserenhet	Ta emot alarm	Ta emot larm status.	X
Rörelserenhet	Mätavstånd	Mät avstånd och larma ifall något detekteras.	X
Rörelserenhet	Vibrationssensor	Larma om vibration detekteras i närhet.	X
Rörelserenhet	Sensor Init	Initierar rörelseenheten.	X
Rörelserenhet	Set Variation	Ger variation på avståndsmätare.	X
Rörelserenhet	Akivering via CAN	Aktiverar larm hantering.	X
Rörelserenhet	Sensor status	aktivering och deaktivering av sensorer.	X
Rörelserenhet	Kalibrering	Kalibrera avståndsmätare via CAN.	X
Rörelserenhet	Can_ID	Ger id till rörelseenhet.	X
Dörrenhet	PING	Skicka ping till centralenhet.	X
Dörrenhet	USART Debug	Debugga dörrenhetens funktionaliteter via usart.	X
Dörrenhet	USART setup	Setup USART.	X
Dörrenhet	Skicka Alarm	Skicka alarm om dörr varit öppen under lång period.	X
Dörrenhet	Ta emot alarm	Ta emot larm status.	X
Dörrenhet	Door Setup	Initialiserar dörrenheten vad antal dörrar, lokala timer och globala timer.	X
Dörrenhet	SET_DOOR_STATUS	Ändra specifik dörr status om den ska vara larmad eller inte.	X
Dörrenhet	Kontrollera dörr status	Ge lokalt alarm då minst en dörr varit öppen under kort tid och skickar globalt alarm efter en längre tid.	X
Dörrenhet	Can_ID	Ger id till dörrenhet.	X
Centralenhet	USART Setup	Setup USART.	X
Centralenhet	USART funktionshantering	Usart funktioner för konfiguration av periferienheter.	X
Centralenhet	Central alarm	Ljud uppspelning via usart och python. Globalt alarm aktiv.	X
Centralenhet	Keypad	Keypad för avlarmning.	X
Centralenhet	Ta emot ping	Larma om enhet inte är ansluten.	X
Centralenhet	Ta emot alarm status	Ta emot alarm status.	X
Centralenhet	Lagra information om periferienheter	Sparar information om periferienheterna i en lista.	X
Störenhet	USART setup	Setup USART.	X
Störenhet	Test av störenhet	Skickar slumpmässig data via CAN bus.	X

C Tabell av tester och resultat

Tabell 1: Tester och dess resultat samt analys

Test	Resultat och analys
Bilaga D Test av knappsats modulen	Testet visade knappsats modulen fungerade som den ska göra, men på grund av glapp eller dylikt i den fysiska knappsatsen hände det ibland att den gav en av signal och sedan en på signal till då tangenten lyftes vilket gjorde att två keypresses detekterades.
Bilaga E: Test av read_code modulen	Testet visade som förväntat att när man skriver in 1234 så lyste det 01 på bargraphen, och när man skrev in ett antal andra fyrsiffriga koder så lyste det 10
Bilaga F: Test av kalibrationen hos vibrations-sensor	Testet visade som förväntat att en för känslig inställning av sensorn resulterade i att den konstant detekterade vibration medan en för okänslig inställning aldrig signalerade för vibration
Bilaga G: Test av kommunikation mellan två MD407 kort via CAN- buss	Testet visade att perienheten kunde skicka meddelande genom CAN bussen och centralenhet kunde ta emot meddelande och detektera att meddelande var ett alarm och tända lysdiod. Detta följde de förväntningar som fanns.
Bilaga H :Test av ping meddelande från var-dera periferienhet till centralenhet	Testet visade att ping funktionen fungerade som planerat men det kan i åtanke förekomma problem vid kombination av andra meddelanden i CAN-bussen
Bilaga J: Test av UltraljudSensor - precision av uppmätning	Testet visade att ultraljudssensorn var funktionell och producerade realistiska resultat enligt de förväntningar som fanns
Bilaga I: Test för initiering av Rörelseenhet via CAN	Testet visade att samtliga funktioner i tabell 1 under bilaga J, fungerade som förväntat
Bilaga K: Test av Störenheten	Testet visade att störenheten fungerade som förväntat, men det finns ett antal problem som kan uppstå
Bilaga L: Test av sensorsystemets usart-state	Testet visade att de funktioner i tabell 1 under bilaga L, fungerade som förväntat
Bilaga M: Test av prototyp	Testet visade att prototypen fungerade som förväntat

D Test av knappsats modulen

Författare : Hugo Söderbergh

Datum : 23-09-05

Specifikation - förväntad egenskap

Knappsats modulen ska hantera drivrutinen för kommunikation via knappsatsen, dvs ge ett interface som tillåter att läsa av när en tangent är nedtryckt eller nyligen nedtryckt. För att åstadkomma detta kommer man först kalla `keypad_init` med vilken port och vilka bits knappsatsen är på, och sedan lägga in `keypad_poll_keys` i sin main loop. Man kan sedan läsa av om en tangent är nedtryckt genom att kalla `keypad_get_keys` för att få ett bitfält med alla tangenter eller `keypad_get_keys_pressed_down` för att få ett bitfält med alla som nyligen trycktes ned.

Metod - Utförande av test

För att testa detta kopplas en knappsats fysiskt till hårdvaran och även en 7-segmentsdisplay för att kunna skicka ut data. GPIO port D låga bitar initialiseras för utmatning till 7-segmentsdisplayen och knappsatsen initialiseras till de höga bitarna. I main-loopen kallar vi `keypad_poll_keys` och gör kod som togglar bit 0 på 7-segmentsdisplayen om det märks att 1 just var nedtryckt, och håller bit 1 hög då D är nedtryckt och låg då den inte är det. Om knappsats modulen fungerar som den ska borde koden togglar bit 0 på 7-segmentsdisplayen då 1 är nedtryckt, och lysa upp bit 1 medan D är nedtryckt.

Resultat

Resultatet blev att knappen D funkade så den skulle utan problem, men när man tryckte på 1 hände det slumpmässigt att lampan togglades en extra gång då knappen lyftes. Testet kördes även för säkerhets skull genom simulatorn, för att kolla om samma problem skedde med toggle knappen. Resultatet där blev att den fungerade så det var tänkt, utan några problem. För att felsöka testades att lägga in en fördröjning mellan varje pollning, vilket tycktes fixa problemet.

Analys

Det verkar som att koden gör det den ska göra, men på grund av glapp eller dylikt i den fysiska knappsatsen händer det ibland att den ger en av signal och sedan en på signal till då tangenten lyfts vilket gör att två keypresses detekteras. Detta kan lösas genom att begränsa hastigheten som knappsatsen läses av vilket skulle kunna göras genom att lägga pollningen som ett avbrott till en timer som ger regelbundna interrupts. En annan enklare åtgärd skulle vara att ha mer saker som händer i ens main-loop då detta skulle begränsa hastigheten som pollningen kan ske vilket skulle fixa problemet.

E Test av read_code modulen

Författare : Hugo Söderbergh

Datum : 2023-09-20

Specifikation - förväntad egenskap

read_code modulen ska kunna göra bakgrundsarbetet för att läsa in en kod upp till 64 siffror lång från systemets knappsats. Först i processen kallar användaren `read_code_initialize` med en null-terminerad sträng av siffror som är den rätta koden. Därefter kallas `read_code_process` varje main loop iteration som ger tillbaka läs-statusen. För att börja läsa in en kod används `read_code_begin` vilket sätter systemet i läs-läge. read_code funktionerna använder sig direkt av keypad.h headern och kräver att knappsatsen initialiseras ordentligt för att fungera. Efter att `read_code_begin` kallats kommer statusen som ges vara `RC_PROCESSING` tills alla siffror knappats in, varefter den kommer returnera `RC_CORRECT` eller `RC_INCORRECT` beroende på om rätt kod knappats in.

Metod - Utförande av test

Testet har bestått av att göra ett testprogram som körs på MD407, som initialiserar knappsatsen med keypad-drivrutinen och ger utsignal till två pins på GPIO D. read_code systemet initialiseras med koden "1234", och skriver 01 om rätt kod skrivits in eller 10 om fel kod skrivits in. Så fort read_code väntar börjar den läsa in en ny kod igen. Allt detta görs genom en oändlig loop som kallar funktionerna enligt specifikationen ovan. Lyckat resultat är att när koden 1234 slås in lyser det 01 på en bargraph som kopplas till GPIO D och om någon annan kombination av 4 siffror slås in lyser det istället 10. Testet är utfört genom simulatorn simserver istället för hårdvara då koden som testas inte är direkt beroende på hårdvaran.

Resultat

Resultatet är att när man skriver in 1234 så lyser det 01 på bargraphen, och när man skriver in ett antal andra fyrsiffriga koder lyser det 10.

Analys

Resultaten stämmer med vad som skulle hända, vilket indikerar att modulen fungerar som den ska.

F Test av kalibreringen hos vibrationssensorn

Författare : Albin Essman, Kevin Cheng

Datum : 2023-09-20

Specifikation - förväntad egenskap

Vibrationssensor ska testas för olika nivåer av känslighet. Känsligheten kan ställas manuellt in direkt på enheten och är essentiell i att få sensorn att fungera korrekt. Felaktig inställning av känsligheten kan resultera i att sensorn konstant detekterar vibration eller att den aldrig detekterar vibrationen oavsett hur stark.

Metod - Utförande av test

Testet inleds med att initiera pin0 på GPIO port D för användning av vibrationssensorn. Detta görs genom att sätta pin0 i input läge med ett pull-up motstånd. En lysdiod kopplas även in för att förtydliga när en vibration detekteras. För att få sensorn att simulera en vibration slogs den lätt till med ett finger.

Resultat

Testet visade att en för känslig inställning av sensorn resulterade i att den konstant detekterade vibration medan en för okänslig aldrig signalerade för vibration.

Analys

Då en del av de möjliga känslighetsnivåerna på vibrationssensorn är oanvändbara så resulterar detta i att inställningen måste ske inom ett mindre intervall vilket komplicerar precisionen.

G Test av kommunikation mellan två MD407 kort via CAN-buss

Författare : Kevin Cheng

Datum : 2023-09-29

Specifikation - förväntad egenskap

Det här testet testar funktionaliteten bakom hur meddelanden mellan två MD407 kort går till. De två MD407 korten som används är centralenheten samt en periferienhet. Denna periferienhet ska endast skicka ett alarm meddelande till centralenheten. För att testa drivrutinerna bakom meddelanden ska ett larm skickas från periferienheten till centralenheten som ska alstra ett larm genom att lysa en lysdiod. Under testet ska ett meddelande föras genom CAN1 porten till endast centralenheten som har sin drivrutin för att bryta ner meddelandet. När centralenheten identifierar meddelandet som ett globalt larm ska global larm tändas.

Metod - Utförande av test

Två MD407 ska vara enheterna som kommunicerar med varandra. RJ-11 kabel ska kopplas mellan de i CAN1 på vardera port. Koden till centralenhet och perienhet ska laddas in i respektive MD407. Därefter ska en lysdiod kopplas centralsystem och slutligen slutligen ska programmet köras.

Resultat

Genomförandet var lyckad då perienheten kunde skicka meddelande genom CAN bussen och centralenhet kunde ta emot meddelande och detektera att meddelande var ett alarm och tända lysdiod.

Analys

Alarm meddelanden har ett specifikt funktions id:n som är unik för alarm. Denna struktur på meddelandet kan implementeras på andra typer av meddelanden som till exempel ping och konfiguration men med sina egna unika funktions id:n.

H Test av ping meddelande från vardera periferienhet till centralenhet

Författare : Kevin Cheng

Datum : 2023-09-29

Specifikation - förväntad egenskap

Både dörrlarms- och sensorenhet ska skicka ping signaler till centralenheten varje sekund. Anledningen är för att kontrollera att CAN bussen för periferienheterna är kopplad till centralenheten. Periferienheterna har timers som skickar ping meddelanden varje sekund genom CAN1. Centralenheten tar emot ping meddelanden och ska kontrollera hur lång tid det tar mellan varje ping intervall. I koden ska ett förbestämd tid för hur länge centralenheten ska ta emot ping innan larmet ska gå igång. För det här testet ska intervallet vara 2 sekunder. Centralenheten använder som periferienheterna en timer som kontrollerar hur lång tid det tar mellan varje ping. Timer filen har funktionen `timer_get_time` som returnerar nuvarande tid efter timern har initialiserats. Tiden mäts i millisekunder och räknas genom: `ping_time_now - ping_time_before >= tids_interval`

Metod - Utförande av test

Dörrenheten och Sensorenheten har en timer som skickar ping signaler varje sekund genom CAN1 till Centralenheten. Centralenheten är kopplad med en RJ-11 kabel som genom en splitter ansluter till både Dörrenhet och Sensorenhet i CAN1. När Centralenheten har detecterat ett ping meddelande sparas tiden. Om tiden mellan ping meddelanden överskrider 2 sekunder slås alarmet på. För just detta test ska centralsystemet larma genom att slå på en pin och lysa en lysdiod.

Resultat

Genomförandet var lyckad. Centralalarmet tog emot ping meddelanden utan problem genom CAN och uppdaterade dess klocka. Sedan ifall man drog ut RJ-11 kabeln slogs alarmet på efter 2 sekunder.

Analys

Ping funktionen fungerade som planerat men det kan i åtanke förekomma problem vid kombination av andra meddelanden i bussen. Det som skulle kunna hända är ifall CAN bussen överflödas med meddelanden med högre prioritet än ping meddelanden. Ping alarmet har i nuläget ingen hög prioritet vilket gör att den är benägen till span som då kommer utlösa alarmet. Dessutom kan 2 sekunder mellan ping intervall vara för kort stund tills alarm.

I Test av UltraljudSensor - precision av uppmätning

Författare : Caesar Paulsson

Datum : 2023-09-27

Specifikation - förväntad egenskap

Målet med detta test är att undersöka precision kring mätningar, för att se vilka förväntningar som kan finnas kring känslighet och maximala avstånd. Dessa känsligheter appliceras genom att sensorns mätvärden jämförs med dess första mätning, och om värdet är utanför ett visst intervall skall larm aktiveras. Intervallet visas nedan:

$$[\text{variation} \pm (\text{variation} * \text{känslighet})]$$

Positivt resultat för undersökningen är att hitta en känslighet där mätresultat samlas utan falska alarm.

Metod - Utförande av test

Genom upprepade försök vid olika längder kan en rimlig känslighet väljas. Testet genomförs inledningsvis med tre olika känslighetsnivåer: 10%, 5%, och 2%. För att undersöka dessa känsligheter väljer vi flera olika avstånd att undersöka, under en tidsperiod av 1 minut. Varje känslighet får tre försök att mäta utan falska alarm, med justeringar i vinkel och höjd för optimering av mätmiljö mellan misslyckade försök. Vid lyckat försök testas även känsligheten genom ett försök att passera förbi. Om sensorn klarar detta räknas detta som ett lyckat test.

Mätningar av avstånd för uppställning av sensor utförs med ett måttband, där sensorn placeras ett visst avstånd från en vägg, utan hinder mellan sensorn och väggen. För ett optimalt mätresultat ska inga objekt heller finnas till sidorna av sensorn, vilket kan då leda till ojämna mätningar.

Resultat

Testet utfördes två gånger, i två olika miljöer. Första försöket utfördes i en miljö utan närliggande hinder, där sensorn placerades på en låda ovanpå ett bord, med inga hinder mellan sensorn och ytan den placerads framför. Fysiska gränser på testmiljön gjorde även att miljön blev mindre optimal vid större avstånd, vilket blev tydligast vid tester runt 4m.

Det andra försöket genomfördes i en sämre miljö, där sensorn placerades likt det första på en låda ovanpå ett bord, men i detta försök var sensorn nära en vägg med hyllor, samt bordet var fyllt med skärmar och diverse objekt, som inte var direkt framför sensorn, men som orsakade falska alarm.

Resultatet för Försöken visas i tabellerna nedan, där siffran visar hur många försök som krävdes, samt ett x, där mätningar inte uppfyllde kraven inom tre försök.

Tabell 1: Resultat från test 1 i optimal miljö

Avstånd	Uppmätt avstånd (kalibrerat i sensorn)	2%	5%	10%
0.5m	0.50m	1	1	1
1m	0.98m	1	1	1
2m	1.99m	1	1	1
3m	3.01m	1	1	1
3.5	3.49m	1	1	1
4m	3.97m	x	x	x(passerades)

Tabell 2: Resultat från test 2 i sämre miljö

Avstånd	2%	5%	10%
0.5m	1	1	1
1m	1	1	1
2m	x	1	1
3m	x	3	2
3.5	x	x	3
4m	x	x	x

Analys

Resultaten visar tydligt sensorns begränsningar i olika miljöer, där värdet på vår variation testas olika. Detta visar att för en miljö perfekt för sensorn kan en låg känslighet på 2-5% vara praktisk, men i sämre miljöer kan sensorn behöva en sämre känslighet, vilket också begränsar längden den kan vara funktionell på, då test 1 visade att 10% gick att passera obemärkt vid ett avstånd på 4m, genom att långsamt följa väggen, vilket är rimligt då vi får en variation på 40cm.

Testet visade på att sensorn är funktionell och producerar realistiska resultat enligt de förväntningar som fanns. Möjliga utvecklingar som kan undersökas är borttagandet av en övre gräns vid intervall, vilket förhindrar felaktiga mätresultat från att skapa falskt alarm, om vi vet att objektet vi observerar inte kommer röra sig.

J Test för initiering av Rörelseenhet via CAN

Författare : Caesar Paulsson

Datum : 2023-10-18

Specifikation - förväntad egenskap

Rörelseenheten initieras genom CAN-bussen, och aktiveras genom funktioner som kallas via USART-ansluten datorn vid centralenheten. Centralenheten ska kunna aktivera individuella sensorer på rörelseenheten, som då och endast då ska kunna skicka larm och aktivera sitt interna larm-state. Rörelseenheten ska vid mottaget avvarningsmeddelande gå ut ur sitt interna larm-state och då återaktivera de sensorer som tidigare varit aktiverade via CAN. Även konfiguration av rörelsesensorn med hjälp av en kalibreringsfunktion samt en funktion för justering av känslighet ska förekomma.

Metod - Utförande av test

För verifikation av denna egenskap görs ett test, där samtliga konfigureringsfunktioner testas vid korrekt och metodisk användning, samt vid felaktig användning, för att hitta oväntade och oönskade beteenden.

Testet börjas genom initiering och aktivering av sensor, där inget larm ska kunna skickas innan sensorerna aktiverats och enhetens CAN-läge initierats. När sensorerna aktiverats testas dess förmåga att skicka larm till centralenheten, som hanterar och skriver av larmet, vilket rörelseenheten ska upptäcka och återigen kunna skicka larm. För att garantera att sensorerna stängs av korrekt deaktiveras de separat och försök till att trigga larmet görs. Slutligen verifieras de två funktionerna specifika för rörelsesensorn, kalibreringsfunktionen och funktionen för justering av känslighet. kalibrering testas genom att rörelsesensorn stängs av, en hand placeras framför sensorn, kalibreringskommandot körs, och sensorn sätts på igen, vilket då ska larm endast då handen tas bort. Verifikation av känslighetens förändring via CAN görs genom att rörelsesensorn tillåts skriva ut sitt nuvarande mätvärden via USART ansluten direkt till enheten. Detta gör att vi kan se exakt de värden som uppmätts och ett platt föremål kan placeras framför sensorn och förflyttas i djupled tills larm triggas, vilket gör att vi kan se direkt om känsligheten observerad matchar den vi ställt in via kommandot i centralenheten.

de funktioner som ska testas är de följande:

Tabell 1: Initering av rörelseenhet

Funktioner	Beskriving	Parametervärde	Parametervärde	Parametervärde
sensor_init	initiera CAN-kommunikation	id	saknas	saknas
motion_on	aktivera båda sensorer	id	saknas	saknas
motion_off	deaktivera båda sensorer	id	saknas	saknas
set_status	individuell kontroll	id	status för sensor: (1/0) = (på/av)	status för sensor: (1/0) = (på/av)
calibrate	kalibrera avståndsmätare	id	saknas	saknas
setVariation	bestäm variation för avståndsmätare	id	variation	variationstyp: cm eller procentenheter

Resultat

Testet visar att samtliga funktioner fungerar som förväntat, sensorerna skickar larm endast efter de initierats och aktiverats, och slutar skicka larm om de deaktiveras. Även Calibrate och setVariation fungerar som förväntat, och låter oss justera rörelsensorn från centralenheten.

Analys

Att alla funktioner fungerar som förväntat innebär att inga större modifikationer behöver göras vidare inom CAN eller rörelsenheten. Något som upptäcktes var behovet att kontinuerligt kunna se vårt nuvarande mätvärden, vilket är användbart för debugging men då frekvensen hos mätningar är väldigt hög är denna funktionalitet opassande för en fungerande prototyp. Detta är troligtvis en funktion som bör läggas till vårt USART-debug läge för rörelsesensorn, vilket gör att en möjlig användare som vill veta det kalibrerade avståndet kan få det.

K Test av störenheten

Författare : Kevin Cheng

Datum : 2023-09-29

Specifikation - förväntad egenskap

Störenheten ska agera som en defekt periferienhet som skickar spam meddelanden över CAN bussen. Störenheten behöver initialiserats först innan användning av centralenheten. Efter initiering ska störenheten kontinuerligt skicka ett meddelande över CAN bussen varje tidsintervall. Denna tidsintervall kan ändras genom USART i störenheten samt också vilket meddelande som ska skickas över CAN.

Metod - Utförande av test

Störenheten sammankopplas med centralenheten och någon fungerande periferienhet (sensorenhet eller dörrenhet). Centralenheten och den fungerande periferienheten ska kommuniceras som vanligt. Från centralenheten ska störenheten sedan initieras med hjälp av USART. När störenhet har initieras ska den lämna setup fasen och börja skicka meddelanden över CAN.

Resultat

Störenheten skickar spam meddelanden genom CAN bussen inom korta intervaller. De intervall som testats var 100ms, 10ms och 1ms. Samtidigt som störenheten skickade spam kunde kommunikation mellan centralenheten och sensorenheten ske med alla tre tidsintervaller. Enheterna tog inte emot meddelanden från störenheten på grund av meddelande filter som finns i alla perienheter.

Analys

Störenheten fungerade som den skulle göra men det finns ett antal problem som kan uppstå. Meddelanden som skickas är inte krypterad som innebär att larmsystemet är i risk för replay-attacker. Dessutom finns det inget säkert meddelande protokoll som filtrerar för alarm meddelande då alarm meddelande inte filtreras på samma sätt som vanliga meddelanden.

L Test av sensorsystemets usart-state

Författare : Caesar Paulsson

Datum : 2023-10-18

Specifikation - förväntad egenskap

Sensorsystemet har ett eget state för testning och debug via usart, där centralenhetens funktioner ersätts av en extern dator, som skickar konfiguration via usart, och tar emot larm, precis likt centralenheten.

Metod - Utförande av test

Verifikation av dessa egenskaper görs genom att implementerade usart-funktioner körs. Vi testar även våra funktioner för larmaktivering inom avståndsmätaren och vibrationsmätaren, där korrekt hantering av vårt nuvarande state ska verifieras.

De funktioner som körs liknar de som förekommer hos centralenhetn och används på samma sätt, med undantag från funktionen som ger oss värdet för avståndsensornas kalibrerade avstånd, som är debugspecifik. De Usart-funktioner som testas visas i tabell 1 nedan:

Tabell 1: Usart-funktioner

Funktioner	Beskriving	Parametervärde	Parametervärde
sensor_init	initiera CAN-kommunikation	saknas	saknas
motion_on	aktivera båda sensorer	saknas	saknas
motion_off	deaktivera båda sensorer	saknas	saknas
set_motion_status	aktivera/deaktivera sensorer	status för sensor: (1/0) = (på/av)	saknas
calibrate	kalibrera avståndsmätare	saknas	saknas
setVariation	bestäm variation för avståndsmätare	variation	Type
GetDist	skriv kalibrerat avstånd i USART	saknas	saknas

Resultat

Testet visar att samtliga funktioner i tabell 1 fungerar som förväntat, och vi kan köra periferienheten utan ansluten CAN utan observerade buggar eller problem.

Analys

Resultatet innebär att vi har ett fungerande USART-system, som går att utöka vid behov

M Test av prototyp

Författare : Caesar Paulsson

Datum : 2023-10-25

Specifikation - förväntad egenskap

Detta test syftar till att verifiera funktionalitet hos en sammansatt prototyp, där de olika enheterna har färdigställt all funktionalitet vid förväntat bruk. Testet undersöker om majoriteten av punkter på kravspecifikationen uppföljs, med undantag debug-lägen och störenhetstest.

Metod - Utförande av test

Testet utförs genom att enheterna kopplas upp, med samtliga sensorer samt 2 dörrar och lampor anslutna mellan kopplingsplattan och periferienheterna. Sedan körs funktioner för initiering på enheterna och CAN-kommunikation upprättas. Vi kan då observera om ping och larm fungerar genom att avlägsna en CAN-kabel, vilket ska aktivera larmsignal via USART. När detta observerats återansluts kabeln och vi deaktiverar larmet genom keypad ansluten till centralenheten. Vi testar sedan våra periferienheter genom att köra konfigureringskommandon, och sedan försöka utlösa och deaktivera larm. Slutligen deaktiveras dörrar och sensorer, och vi verifierar att de stängts av korrekt.

Resultat

Resultatet visar att prototypen fungerar som förväntat. Detta innebär att prototypen konfigureras via CAN. Prototypens periferienheten skickar ping-meddelande, samt larmar när sensorer aktiveras, vilket skickar signaler till centralenheten som observeras via ansluten USART genom alarmljud. Centralenheten tar även emot ping-meddelande och kontrollerar att alla anslutna enheter fortsätter vara anslutna. Även deaktivering av enheter och aktivt larm ska verifieras via centralenheter.

För att sammanfatta de olika moment som testats samt resultat används tabell 1 nedan. '

Tabell 1: testresultat för prototyptest

Funktionalitet:	Resultat
Initiering	Fungerar
Ping	Fungerar
Alarm	Fungerar
Avlarmning	Fungerar
Konfigurering	Fungerar
Ljudaktivering	Fungerar
keypad	Fungerar
rörelsesensorer	Fungerar
dörrsensorer	Fungerar

Analys

Genom vårt övergripande test har vi verifierat att ett fungerande system har byggts upp, samt att de individuellt testade enheterna fungerar sammansatta enligt vår specifikation.

N Kommunikation och Samordning

Under projektets utveckling skedde den största delen av kommunikationen på fysiska arbetsmöten, men kommunikationen skedde även virtuellt via en online kommunikationskanal så som en Discord server.

Tidigt i utvecklingsprocessen blev det tydligt att det utnyttjandet av fysiska arbetsmöten var effektivt. Eftersom gruppmedlemmar kunde fråga varandra om hjälp och inte sitta fast på de individuella problem som stöttes på under projektutvecklingen. Dessutom hölls även veckovisa fysiska möten med gruppens mentor där frågor om projektet ställdes och hjälp utnyttjades vid behov. Dessa möten var ett bra sätt att se till att arbetet höll rätt takt och rörde sig framåt. Det sågs även till att aktivt fråga medlemmar i projektet om de behövde hjälp under dessa möten eftersom det inte alltid är lätt att be om hjälp och det problemet var något som ville undvikas.

Utöver gruppmöten arbetades projektet även enskilt då saker som t.ex. läsning om hårdvara och mjukvara fungerade bra själv och även kodning. Om det stöttes på problem med kodningen fanns det inte bara möjlighet att ta upp det på nästa arbetsmöte men även online genom Discord. Ytterligare användes även Discord för planeringen av fysiska möten och upplysning av projekt-deadlines.