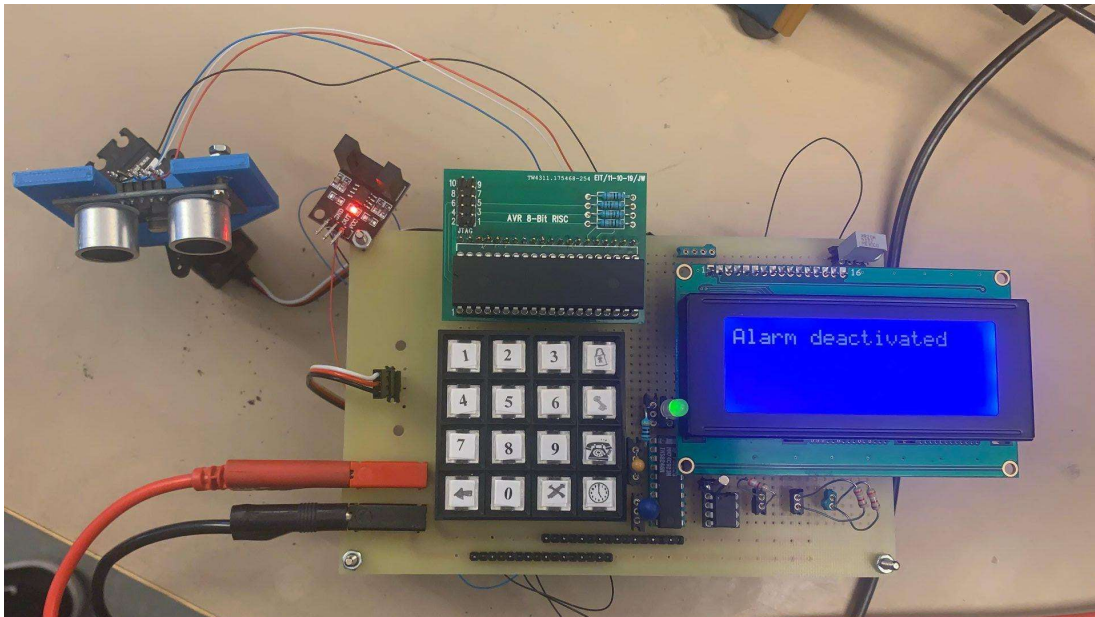


EITF12: Larmsystem

Joakim Dworak, Rickard Gustafsson, Isak Ottosson

2022-05-15



Abstract

The purpose of this report is to describe the “alarm system”-project created by Group 10 in the course EITF12. The main goal of the course was to teach the students about processes in developing systems and devices in an industrial environment.

A long list of components were connected together on a circuit board, according to a previously created electronic schematic, to form the physical alarm system, after which the software development phase of the project started using the programming language C.

The end result of the course is a functioning prototype of an alarm system and a lot of experience for the students in the group.

1. Inledning	3
1.1 Kravspecifikation	3
2. Hårdvara	4
2.1 Processor	4
2.2 LCD-skärm	4
2.3 Knappsats	4
2.4 Encoder	4
2.5 Servo	4
2.6 Ultraljudssensor	5
2.7 Läsgaffel	5
2.8 Lysdiod	5
2.9 RTC	5
2.10 Kristall	5
3. Genomförande	6
3.1 Montering av hårdvara	6
3.2 Utveckling av mjukvara	6
3.3 Beskrivning av källkod	6
4. Kopplingsschema	8
5. Resultat	9
6. Diskussion	10
6.1 Förbättringar	10
6.2 Annan hårdvara	10
6.3 Andra funktioner	10
7. Appendix	11
7.1 Bilaga A	11
7.2 Bilaga B	12
7.3 Bilaga C	13

1. Inledning

Kursen Digitala Projekt - EITF12 är utformad i syftet att studenter ska få lära sig genom praktiskt arbete hur utvecklingsarbete inom digital industri är utformat. Som bas för arbetet tillhandahålls grupper om 2-3 personer med ett ATmega16 kort och nödvändiga komponenter för en egen utformad idé. Denna idé skall sedan tas till ett prototypstadie för att demonstreras ihop med redovisning av tillhörande dokumentation.

Vi valde att bygga ett larmsystem med flera funktioner. Primära larmfunktionerna är att larmsystemet med hjälp av en avståndsmätare mäter runt omkring för att sedan upptäcka ifall något har förändrats. Det finns också en infraröd sensor som skall agera som en dörr. För att hantera larmet smidigt så har vi lagt till en knappsats och en LCD-skärm. För att signalera för användaren i vilket läge larmet befinner sig så lyser en lysdiod antingen rött eller grönt.

1.1 Kravspecifikation

I början av projektets gång sattes följande konkreta krav på den färdiga prototypen:

1. Användaren ska få välja en fyrsiffrig pinkod som sedan också måste matas in en gång till för att bekräftas.
2. På displayen ska användaren få information om larmets status.
3. Som komplement till displayen ska även en lysdiod lysa i olika färger beroende på larmets status.
4. En avståndsmätare ska agera som scanner för att ta reda på om något i rummet har förflyttat sig. Upptäcks att något förflyttat sig ska larmet utlösas. Avståndsmätaren ska rotera på ett servo.
5. En sensor som ska illustrera ett fönster som öppnar sig ska utlösa larmet.

Under projektets gång uppkom även ett nytt krav på larmet, vilket var att larmet skulle kunna ställas in på en timer. Användaren ska alltså kunna specificera en viss tid då larmet ska aktiveras.

2. Hårdvara

Vid konstruerandet av larmet krävdes en del hårdvara som nedan beskrivs kortfattat. Utöver nedanstående komponenter användes dessutom diverse resistorer, kondensatorer och kablage. Ett blockdiagram över hur komponenterna är kopplade kan ses i bilaga A.

2.1 Processor

En 8 bitars-mikrokontroller av typen ATmega16 används för att styra och kontrollera hela systemet. För att kommunicera med processorn från en extern dator användes en Atmel-ICE development tool med gränssnittet JTAG. Detta tillåter enkel testning, debugging och överföring av slutgiltig C-kod till processorn.

2.2 LCD-skärm

För att kommunicera med användaren av larmsystemet användes en LCD-skärm med plats för 4x20 tecken. Denna kontrollerades av en Hitachi 44780-controller.

2.3 Knappsats

En 16-knappars knappsats användes för att ge användaren möjlighet att mata in siffrorna 0-9, samt kunna välja mellan att larma på direkt, larma av eller larma på ett visst klockslag. Övriga knappar lämnades utan någon speciell funktion.

2.4 Encoder

För att kunna tolka de signaler som kommer från knappsatsen användes en MM74C923, vilket är en 20-knappars-encoder. Denna kunde användas till knappsatsen genom att utesluta två av pinsen som annars skulle svara för den femte raden på knappsatsen.

2.5 Servo

Ett Parallax Standard Servo användes för att tillåta avståndsmätaren att rotera till olika positioner. Detta servot kan rotera 180 grader, vilket ger tydligt definierade start- och slutpositioner.

2.6 Ultraljudssensor

För att kunna utföra avståndsmätningar användes en ultraljudssensor av modellen HY-SRF05, vilken placerades på servot. När processorn skickar en signal till ultraljudssensorn skickar denna i sin tur ut en våg av ultraljud och läser av tiden det tar för vågen att komma tillbaka till sensorn. Denna tid förmedlas sedan till processorn och sparas för att kunna jämföra hur lika mätningarna är för varje gång som en avläsning sker.

2.7 Läsgaffel

En gaffelformad IR-sensor av modell LM393 H2010 användes för att illustrera ett fönster som öppnas. Då ett objekt förs in i läsgaffeln och bryter av det infraröda ljuset skickas en signal till processorn, vilket utlöser larmet om det är aktiverat.

2.8 Lysdiod

För att ge användaren av larmet en indikation på larmets status användes en lysdiod som kan lysa grönt eller rött. För att indikera att larmet är avstängt lyser dioden med ett fast grönt ljus, och för att indikera att larmet är aktiverat lyser dioden med ett fast rött ljus. Då larmet blivit utlöst så blinkar dioden rött.

2.9 RTC

För att möjliggöra för användaren att ställa in en tidsbestämt larm krävs att systemet håller koll på tiden. Detta åstadkoms med en RTC-krets av modellen MCP7940M, som använder kommunikationsprotokollet I²C. Denna fungerar som en klocka som går att ställa genom information från processorn för att därefter ticka på varje sekund med hjälp av pulser från en kristall. Den aktuella tiden går sedan att läsa av direkt till processorn.

2.10 Kristall

För att RTCn ska kunna hålla tiden mer exakt användes en kvartskristall som börjar vibrera då den ges en spänningspuls. Den kristallen som användes i vår applikation vibrerar med frekvensen 32,768 kHz, vilket gör det lätt för RTCn att beräkna när en sekund passerat.

3. Genomförande

3.1 Montering av hårdvara

Det första steget i utvecklingsarbetet var att bestämma vilka funktioner som förväntades av larmsystemet. I diskussion med handledare fastställdes sedan vilka komponenter som skulle vara nödvändiga för genomförandet av detta. Blockschemat i bilaga A togs sedan fram och ett preliminärt kopplingsschema producerades. Utifrån denna konstruerades sedan en första version av larmsystemet. Komponenterna som specificeras i avsnitt 2. Hårdvara installerades på en kopplingsplatta och kopplades samman med hjälp av sladdar, resistorer och kondensatorer. Själva kopplingarna gjordes med lödning eller trådbindning.

Inledande testning visade att vissa komponenter behövde bytas ut på grund av defekter eller andra former av inkompatibilitet med resterande delar av systemet. Efter justeringar ritades ett slutligt kopplingsschema, se figur 1.

3.2 Utveckling av mjukvara

Därefter inleddes arbetet med mjukvaruutveckling i programmeringsspråket C med hjälp av programmet Atmel Studio 7. Baserat på information från datablad för varje komponent Under hela utvecklingsprocessen testades systemet kontinuerligt för att se att alla komponenter uppträdde som förväntat. När hela programmeringen färdigställdes testades systemet sedan ytterligare för att eventuella buggar skulle upptäckas.

3.3 Beskrivning av källkod

Då larmet startas ombeds användaren att mata in aktuell tid. Är denna av felaktigt format noteras användaren och ombeds mata in en ny tid. Då en korrekt tid matas in noteras detta i koden. Därefter ombeds användaren att välja en pinkod samt bekräfta denna. Stämmer inte inmatningen andra gången får användaren välja en ny pinkod. Då en korrekt pinkod satts går larmet in i avaktiverat läge.

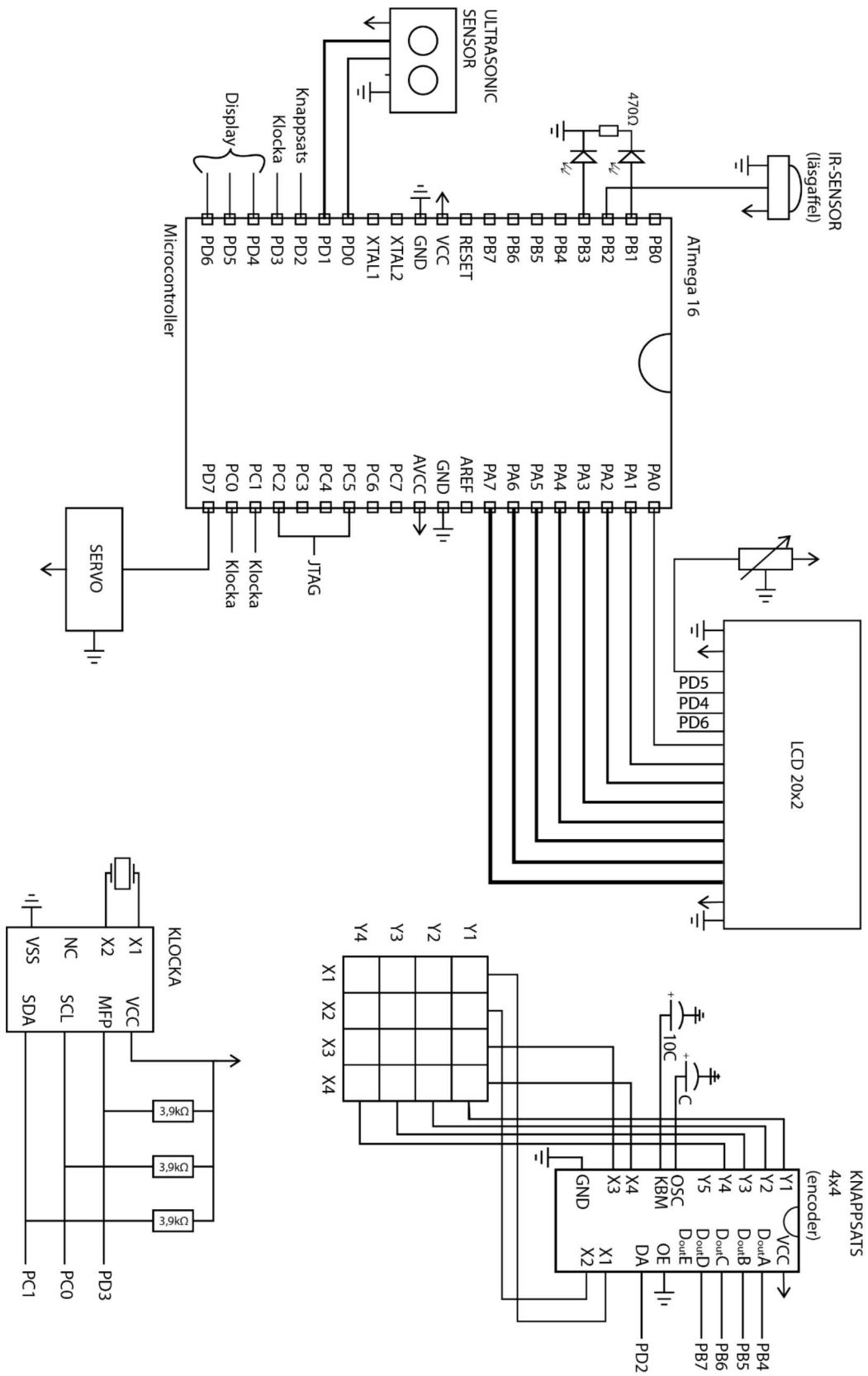
Användaren kan nu välja att antingen starta larmet direkt eller starta larmet vid en viss tidpunkt. Detta görs med två olika knappar på knappsatsen, vilka är de enda knapparna som larmet lyssnar på i avaktiverat läge. Väljs knappen för att larma direkt ombeds användaren att mata in den valda pinkoden. Matas fel kod in meddelas användaren och larmet återgår till avaktiverat läge. Om sensorn för fönstret har vidrörts, vilket simulerar ett öppet fönster, meddelas användaren att man ska stänga det (vilket görs automatiskt), och användaren får trycka på knappen för att larma på igen. Vid korrekt inmatning av pinkod övergår larmet till aktiverat läge och servot med avståndsmätaren gör en första scan av rummet. Mätvärdena kontrolleras sedan kontinuerligt då larmet är aktiverat och servot roterar. Märks en avvikelse eller om sensorn för fönstret vidrörs utlöses larmet direkt.

Väljs istället knappen för att larma på en viss tid ombeds användaren att mata in en korrekt tid. Om detta inte görs meddelas användaren och larmet återgår till avaktiverat läge. Vid korrekt inmatad tid får användaren mata in pinkoden för att aktivera larmet. Görs inte detta sker samma sak som när användaren väljer att aktivera larmet direkt. Vid korrekt inmatning av pinkoden meddelas att larmet är schemalagt och larmet väntar på att rätt tid ska slå in. Därefter sker samma procedur som när användaren aktiverar larmet direkt.

I aktiverat läge lyssnar larmet efter en knapptryckning för att avaktivera larmet. Trycks denna knapp in ombeds användaren att mata in korrekt pinkod. Görs inte detta inom tre försök så utlöses larmet.

Ett flödesdiagram för hur logiken i källkoden fungerar kan ses i bilaga B och C.

4. Kopplingschema



Figur 1: Kopplingschema för larmsystemet.

5. Resultat

Det slutliga larmsystemet som producerades uppfyller alla krav som ställdes upp i början av arbetet och får därför ses som mycket lyckat resultatmässigt. Det enkla användargränssnittet är lätt att förstå och larmet klarar av alla funktioner som förväntas, som att larma på, larma av och utlösas ifall någon av sensorerna aktiveras.

6. Diskussion

6.1 Förbättringar

En tydlig förbättring som kunde ha gjorts är att använda en annan sorts kablage till servot och ultraljudssensorn. De är dels relativt korta, så servot måste vara placerat nära kretskortet med övriga komponenter, men även oflexibla. Detta blir ett problem då ultraljudssensorn, vilken är kopplad till kretskortet, roterar på servot och drar med sig kablarna runt.

6.2 Annan hårdvara

För att kunna förbättra felkänsligheten på larmet skulle en annan typ av ultraljudssensor kunna användas. I dagsläget är sensorn mycket känslig för diverse störningar och kan därmed ge orimliga mätvärden, även i en strikt kontrollerad miljö. Det kan dessutom uppstå felaktigheter i mätvärdena då en mätning sker på ett objekt som står framför ett annat om själva mätningen sker på kanten. Detta blir ett problem då sensorn roterar på servot och mätpunkterna inte alltid tas på exakt samma ställe vid varje rotationscykel. Uppkomsten av detta fenomen minskade dock då servot fastsattes med dubbelhäftande tejp.

En annan komponent som skulle kunna bytas ut för att öka användarvänligheten är knappsatsen, då denna kan tyckas vara lite seg att mata in önskade kommandon med. Det händer att den inte registrerar att en knapp tryckts in, vilket gör att användaren får trycka på samma knapp igen.

6.3 Andra funktioner

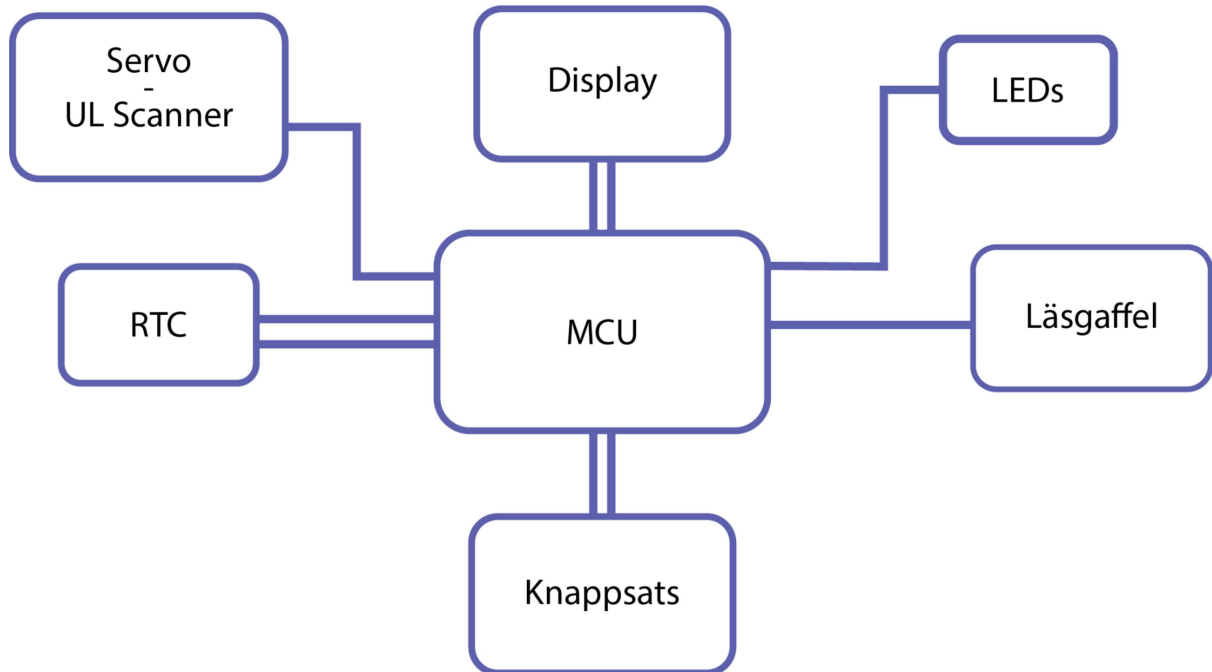
För att ytterligare öka användarvänligheten på larmet så skulle fler funktioner kunna tänkas ha implementeras. En idé hade varit att implementera en backstegsknapp för att låta användaren ta bort en felaktigt inmatad siffra. I dagsläget måste användaren med flit mata in fel kod om man vet med sig att man matat in fel siffra för att försöka igen.

En annan funktion som skulle kunna implementeras är att byta pinkod då larmet är inaktiverat. För att göra detta i dagsläget måste strömmen brytas för att på så sätt starta om larmet helt.

Slutligen skulle även klocksystemet kunnat ha fler funktioner. Bland annat så skulle det kunna vara fördelaktigt att kunna byta tid medan larmet är på, istället för att likt med pinkoden behöva starta om larmet. Även en funktion att kunna visa den nuvarande tiden hade varit fördelaktigt.

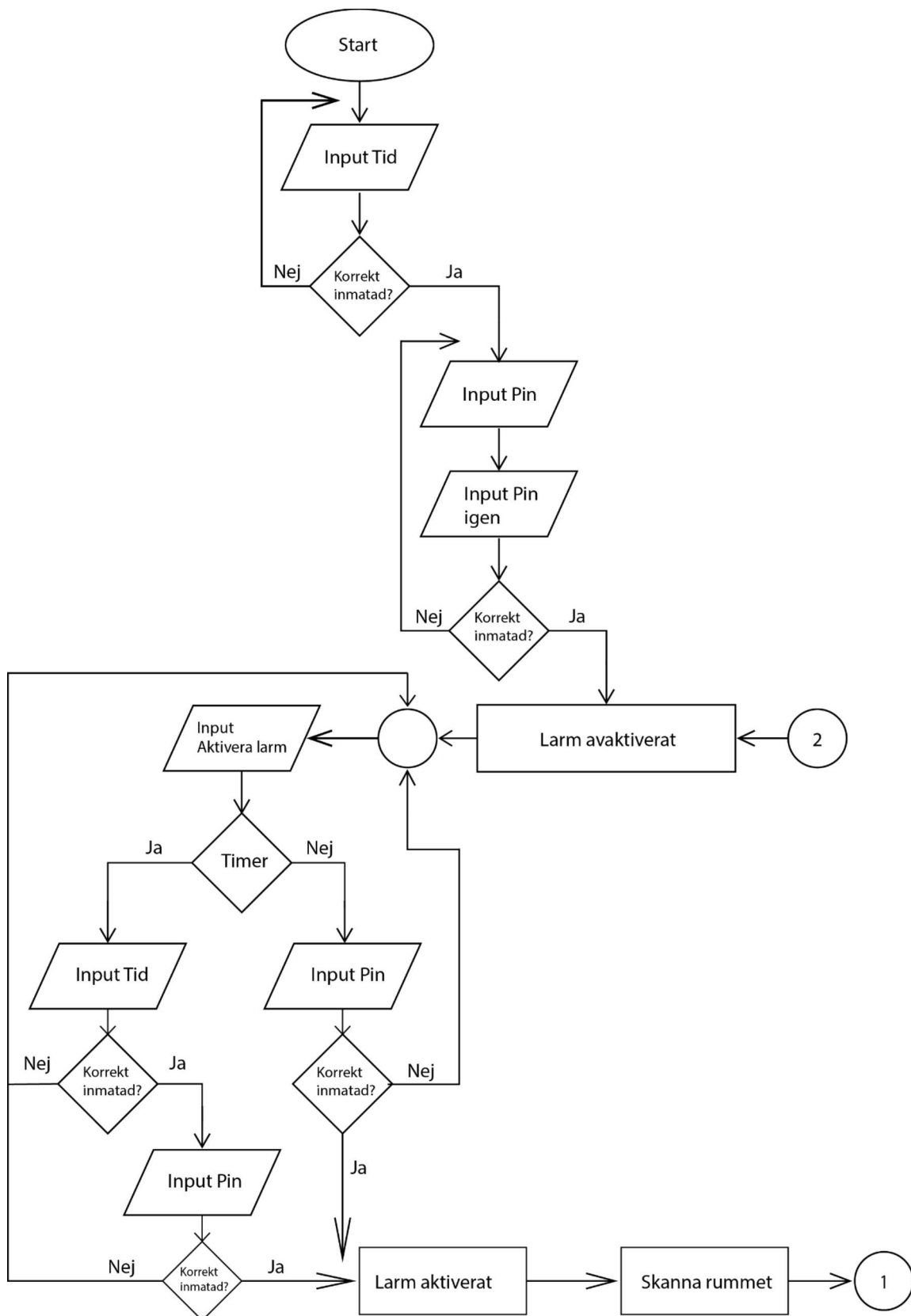
7. Appendix

7.1 Bilaga A



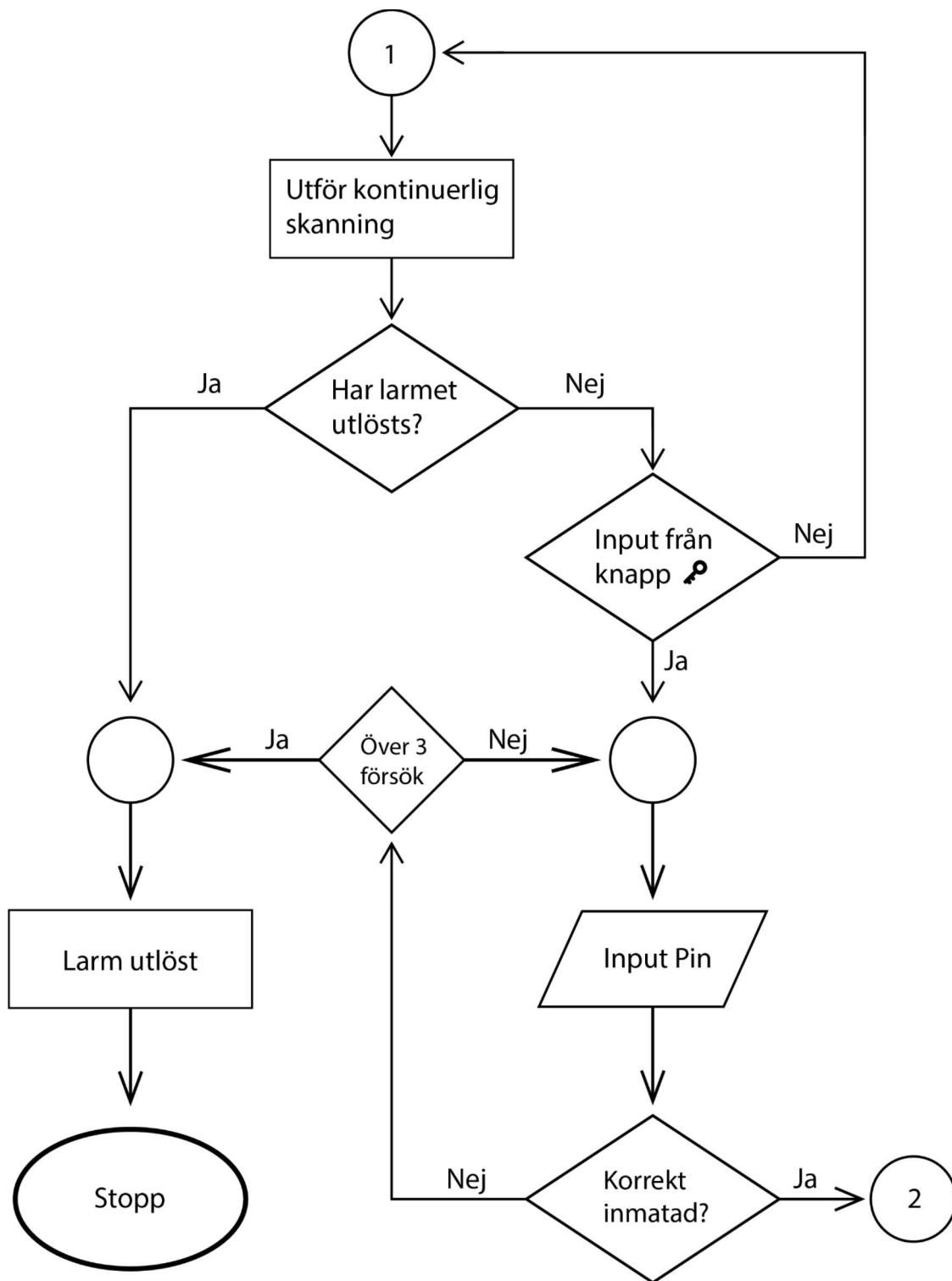
Bilaga A: Blockdiagram för hur larmsystemet är kopplat.

7.2 Bilaga B



Bilaga B: Flödesdiagram över källkoden för larmet.

7.3 Bilaga C



Bilaga C: Flödesdiagram över källkoden för larmet.