

Kamerasystem för detektering av skadegörelser vid busshållplatser



Yurdaer Dalkic
George Albert Florea
Louay Khalil
Benjamin Sejdic

19 February, 2017 Malmö

Abstract

Rapporten beskriver ett kamerasystem där en Axis-övervakningskamera ska styras med hjälp av ett par sensorer tillsammans med ett trådlöst inbyggt system, för att övervaka utsatta busshållplatser. Syftet med detta system är att upptäcka skadegörelser utan att kameran ska filma dygnet runt helt i onödan, samt att skapa mer trygghet. En övergripande beskrivning av systemets användning ingår i rapporten tillsammans med en beskrivning av de olika delarna.



Figur 1: Gruppen SAFE24 (www.gaia3d.co.uk/about/ redigerad)

Innehåll

1	Teori	2
1.1	Problemlösning	2
2	Material & Metoder	3
2.1	Material	3
2.2	Metoder	5
2.3	Systembeskrivning	6
2.4	Tekniska Aspekter	6
2.5	Arbetsuppgifter	7
3	Resultat	8
4	Lag och Etik	9
	Litteraturförteckning	10
5	Bilaga	11
5.1	GitHub-länk	11
5.2	Diagram	11
5.3	Testfall	15

1

Teori

1.1 Problemlösning

En busshållplats som är övervakad dygnet runt är en ineffektiv lösning. Övervakningen ska ske i samband med att specifika villkor är uppfyllda. Därför är glasen monterade med en varsin ljudsensor för att registrera antingen hårda slag mot glasytan eller även när glasen krossas. Dessa två händelser utgör ett huvudvillkor för att systemet ska aktiveras. Om inget villkor är uppfyllt kommer systemet att vara i ett passivt tillstånd och enbart lyssna på förändringar. Utöver det finnas en PIR-sensor som detekterar rörelser inne i hållplatsen. Om rörelse detekteras tänds en lampa för att ge en känsla av trygghet för dem som väntar inne i hållplatsen, men även för att uppmärksamma bussarna och andra i omgivningen om att någon rör sig i hållplatsen.

I mitten av vägen ska det finnas en IP-kamera. När övervakningen börjar så riktar sig IP-kameran först mot den ljudsensor som har registrerat en händelse för att filma i den riktningen ett antal sekunder innan den försätter filmandet medan den roterar 360 grader tillbaka till den första punkten. Där filmar den ytterligare några sekunder till i samma riktning. Därefter slutar kameran att filma och återgår till ett passivt läge.



Figur 1.1: En obevakad hållplats i Malmös stadsdel Lindängen. (Google Maps: Lindängen - Malmö)

2

Material & Metoder

2.1 Material

I vår lösning använde vi oss av ett utvecklingskort ESP8266 med ett inbyggt Wifi som var nödvändig för att kunna kommunicera med det nätverk som kameran var uppkopplad till.

Kameran är av modellen Q6128-E Network Camera med möjlighet till internetuppkoppling. Upplösningen som används är 3840x2860. Ett suffix med datum och tidsinformation läggs till i filnamnet för inspelningen.

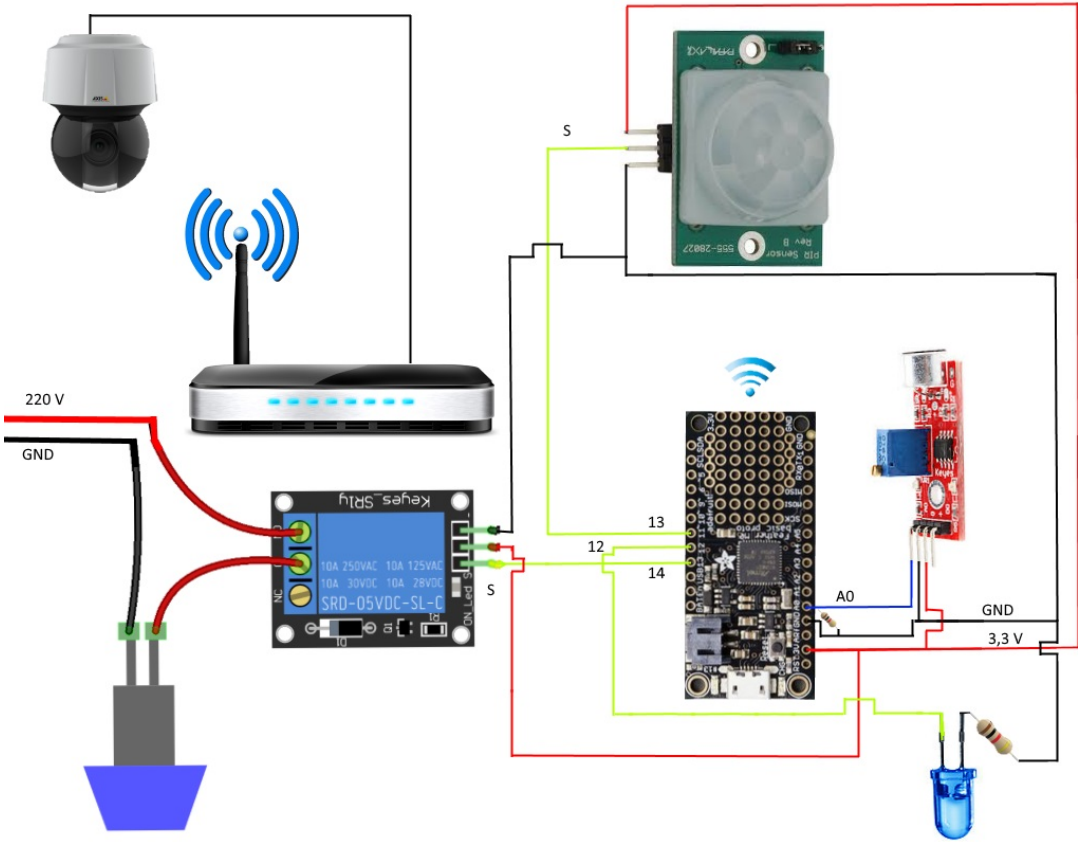
En PIR-sensor från Adafruit användes som rörelsedetektor för att tända lampan i hållplatsen.

Som nämt tidigare i rapporten använde vi oss av en ljudsensor som var monterad mot en glasyta. Vid besök hos en elektrokit-grossist kunde vi inte få tag på någon användbar trycksensor som var tillräckligt känslig för att ersätta ljudsensorn. Under demo-dagen fick vi frågan om vi hade tänkt på en Piezo-sensor, som är en typ av en tryck-sensor. Efter fakta-sökning fann vi att en Piezo-sensor hade varit ett bättre alternativ än ljudsensorn.

För att kunna identifiera att någon har utfört vandalisering måste systemet lagra bilder/inspelningar på en server. En FTP-server användes. FTP-servern och IP-kameran låg i samma subnät.

Information om FTP-servern:

- IP adress : 192.168.0.106
- Port nummer : 21
- Användarnamn : "FTP-User"
- Lösenord : "Safe24"



Figur 2.1: Alla komponenter kopplade med varandra.)

2.2 Metoder

Om PIR-sensorn detekterar någon rörelse då skall lampan tändas en viss tid och sedan släckas. Om lampan är tänd och PIR-sensorn detekterar ny rörelse då fortsätter lampan vara tänd.

På grund av brist på glas och utrymme för oss att göra testförsök med att ta sönder glas använde vi istället plast. Plasten skulle föreställa glaset i en busshållplats. Ljudsensorn är precis som PIR-sensorn också igång hela tiden. Om ljudsensorn detekterar ett ljud som är över gränsvärdet skall den viktiga processen börja. Tester vi utförde på systemt för att filtrera bort störningar medförde att ljudsensorn fick ett tröskelvärde på 75 utav max 1024 som den analoga ingången kunde läsa. Över denna gräns indikerades att glas hade gått sönder. Eftersom denna enkla signalbehandlingen av ljudsensorn räckte för att uppnå våra önskemål och lösa uppgiften så krävdes ingen mer avancerad filtrering av signalen.

IP-kameran var installerad i mitten av vägen så att den kan rotera fritt eftersom vandaliseringen kan ske på avstånd från hållplatsen. Den var ansluten via ethernet och vi styrde den med hjälp av http-kommando som utvecklingskortet skickade iväg. Vi fick kamerans http-API (VAPIX) från Axis. Inne i kameran skapade vi tre events ActionPTZStation1, ActionRecord, ActionPTZHome.

ActionPTZStation1: När virtuell port 8 aktiveras så riktas kameran mot en bestämd position som heter plats1” (busshållplatsen).

ActionRecord: När virtuell port 9 aktiveras så börjar kameran videoinspelningen. Efter avslutad inspelning skickas klippet till FTP-servern.

ActionPTZHome: När virtuell port 10 aktiveras så riktas kameran mot en bestämd position som heter Safe24 och som motsvarar start och slutpositionen för kameran.

2.3 Systembeskrivning

Vi har nämnt tidigare i rapporten att vårt kamerasystem skall befinna sig i ett passivt läge så länge inget villkor är uppfyllt. Om ett villkor är uppfyllt då aktiveras två virtuella portar i kameran. Den ena riktar kameran mot busshållplatsen (port 8) och den andra (port 9) gör att kameran börjar filma. Då filmar kameran i fem sekunder och sedan filmar den runt omkring medan den vrider sig horisontellt. Efter ett varv riktas kameran mot busshållplatsen igen och det görs genom att aktivera virtuell port 9. Kameran ska försätta filma några sekunder till och efter det ska den riktas mot sitt standardläge dvs ActionPTZHome. Den totala videolängden är 40 sekunder och filmen skickas till FTP-servern. Ifall ljudsensor detekterar ett annat ljud som är över gränsen inom 40 sekunder då ska processen börja från början dvs kameran riktas tillbaka mot bushållplatsen osv. Samtidigt som utvecklingskortet väntar på utslag från ljudsensorn, väntar den på utslag från PIR-sensorn. När någon rör sig inne i hållplatsen ser utvecklingskortet till att en lampa tänds och släcks en viss tidsperiod som hela tiden förlängs vid nya rörelser. I det fall där kamerasystemet aktiveras, tar kamerasystemet över PIR-sensorn och låter lampan att blinka tills kamerasystemet är tillbaka till sitt passiva läge.

2.4 Tekniska Aspekter

Pir-sensorn vi använde oss av gav oftast låg signal när det inte förekom rörelse. I de fall där PIR-sensorn gav hög signal när ingen rörelse förekom har detta åtgärdats genom att filtrera signalen. Med hjälp av några fördröjningar och en räknare skapade vi en såkallad runda. På så sätt kunde vi anpassa signalerna för att detektera enbart rörelse. Räknaren räknade in 7 olika signalvärden. Tester av systemet visade att enbart 7 höga värden i rad motsvarade att rörelse detekterats. Ibland hände det att vi fick höga värden konstant från PIR-sensorn även när vi inte hade rörelse. Denna störning filtrerade vi bort direkt efter en avslutad runda; genom att omvandla utvecklingskortets input-pin till output-pin och därefter skriva in ett lågt värde. Givetvis gjorde vi om output-pin till input-pin innan räknaren skulle börja med en ny runda. När det gällde mikrofonen så behövdes ingen avancerad signalbehandling annan än att vi hade ett tröskelvärde att läsa in värden efter.

Biblioteket som användes för att schemalägga tasks är skapat av Nicholas Wiersma och heter ESP8266Scheduler. Schemalägningsalgoritmen som biblioteket använder är enkel att implementera men saknar viktiga funktioner som exempelvis prioritering; vi hade ingen möjlighet att påverka prioriteringen. Med hjälp av detta bibliotek skapade vi tre olika tasks, en för nätverksuppkopplingen (WifiTask), en för PIR-sensorn (pirTask) och en tredje för ljussensorn och kameran (micTask). Två av de tre tasken (pirTask och micTask) agerar som inputs till systemet. Den tredje tasken kontrollerar om det finns en WiFi-anslutning, om anslutningen är nere försöker den återansluta till wifi.

Vi visste att när alla tasks hade initierats så kunde varje task hoppa över till en ny

task efter en viss okänd tid. Att tiden var okänd betraktade vi som oväsentligt eftersom den tasken som hade hand om glaskrossningen och kamera funktionerna utförde sina uppdrag i enlighet med våra önskemål. Detta hade vi fått bekräftat genom våra tester av systemet.

Vi visste att delayerna inne i en task styrde funktionerna i tasken. Däremot såg vi från våra tester att en och samma task kunde köras två gånger i rad utan att delayerna i föregående task påverkade. Det saknades information om vilken typ av schemaläggning som används i biblioteket som vi använde för schemaläggningen. Utifrån våra tester så verkar tasksen turas om beroende på delayerna som finns inne i varje task.

2.5 Arbetsuppgifter

Gruppen arbetade både tillsammans och även enskilt så att var och en av gruppmedlemmarna kunde bidra med något. Benjamin var delaktig i arbetet med flödesdiagram, tasks, hjälp med byggandet av busshållplats och även testfall. Yurdaer arbetade med wifi-kodning av ESP:n, kamera-kommandon, ftp-servern och i rapporten skrev han om sina delar samt om etiska aspekter. Georges bidrag var med struktur av kod, API, manual, testfall, FTP-servern och en del rapportskrivning. Han bidrog även med upprättande av en Latex-mall för rapporten. Louay arbetade med sensorernas kodning, ihopkoppling av alla komponenter, förslag till router-lösning, byggandet av en hållplats och kontinuerlig testning av alla kopplingar.

3

Resultat

Under testfasen användes det webbaserade programmet testrail. Fördelen med det var att flera personer kunde lägga till och redigera testfallen samt att vi fick en grafisk överblick över vilka testfall som har passerat respektive fallerat. Testfallen skrevs på så sätt att de skulle validera systemlösningen vilket gick ut på att kontrollera kamerans bestämda rörlighet, sensorernas känslighet, kommunikationen mellan esp8266 och kameran, kommunikationen mellan kameran och ftp-servern samt att testa kamerans anslutning till wifi nätverket. Vi testade schemaläggningen genom att skriva ut ett visst ord i slutet av varje task, och vänta på att se det ordet utskrivet. När vi startade programmet såg vi att vissa task kördes flera gånger innan den hoppade till nästa task, dock kördes alla task i ordning. Resultaten från samtliga testfall överensstämde med det förväntade resultatet. Som en följd av detta passerade alla tester och inga justeringar av systemlösningen var nödvändiga. Med andra ord har systemlösningen validerats att den löser problemet genom testningen.

Lag och Etik

I Sverige finns det regler och lagar som gäller för kameraövervakning. Kameraövervakningslagen (2013:460) omfattar dels övervakningskameror, dels tekniska anordningar för att behandla eller bevara bilder och andra tekniska anordningar för avlyssning eller upptagning av ljud som används i samband med övervakningskameror [1]. Enligt lagen är att tillstånd krävs om:

- kameran riktas mot ”en plats dit allmänheten har tillträde”
- utrustningen kan användas för personbevakning
- kameran är uppsatt utan att manövereras på platsen

I definitionen ”allmänheten har tillträde” tar man ingen hänsyn till om det handlar om privat eller allmän mark, utan alla platser dit allmänheten någon gång har tillträde omfattas av lagen om allmän övervakning. Busshållplatser och gator räknas som allmänna platser enligt definitionen av allmän plats. Det här ställer vissa krav på den som installerar och/eller äger det system som vi har skapat under detta projektet. Man måste se till att lagar och regler följs dvs att man måste göra en ansökan om tillstånd till allmän kameraövervakning.

Kameraövervakning är en metod som används för att minska brottslighet. Effekterna varierar beroende på hur man arbetar med kamerorna. Kameraövervakning är en känslig fråga utifrån ett integritetsperspektiv. Det som människor är oroliga för när det gäller övervakningskameror är att deras integritet kränks. Ingenjörer har ansvar att verka för att tekniken används för samhällets och mänsklighetens bästa enligt Hederskodexen för Sveriges Ingenjörer [2]. Vi använder kameraövervakningen i syfte att bekämpa brott vilket är bra för både samhället och människorna i det samhället.

Litteraturförteckning

[1] <http://www.lansstyrelsen.se/>

[2] <http://www.sverigesingenjorer.se/om-forbundet/sa-tycker-vi/hederskodex/>

[3] <https://www.bra.se/bra/forebygga-brott/kameraovervakning.html>

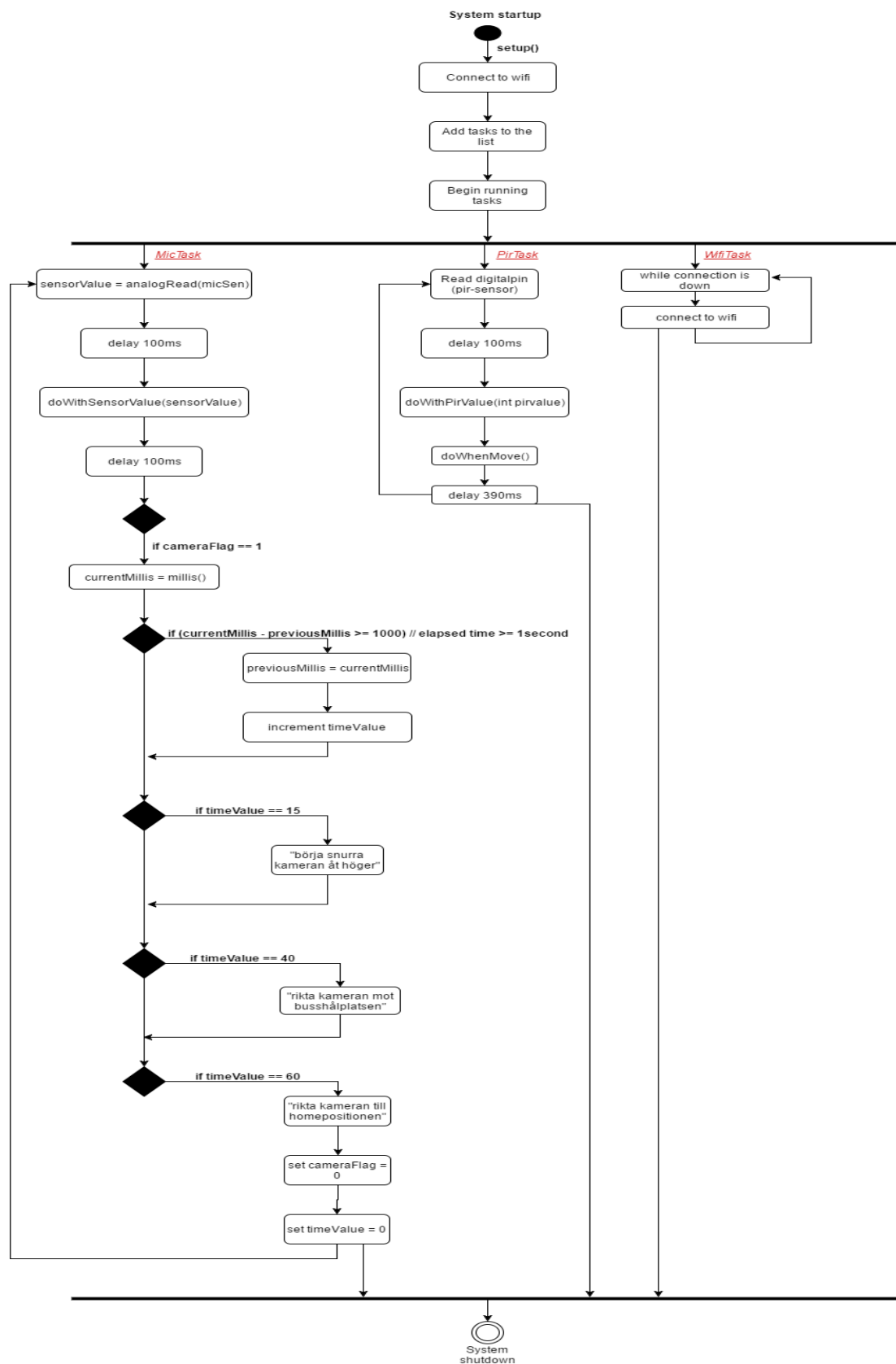
5

Bilaga

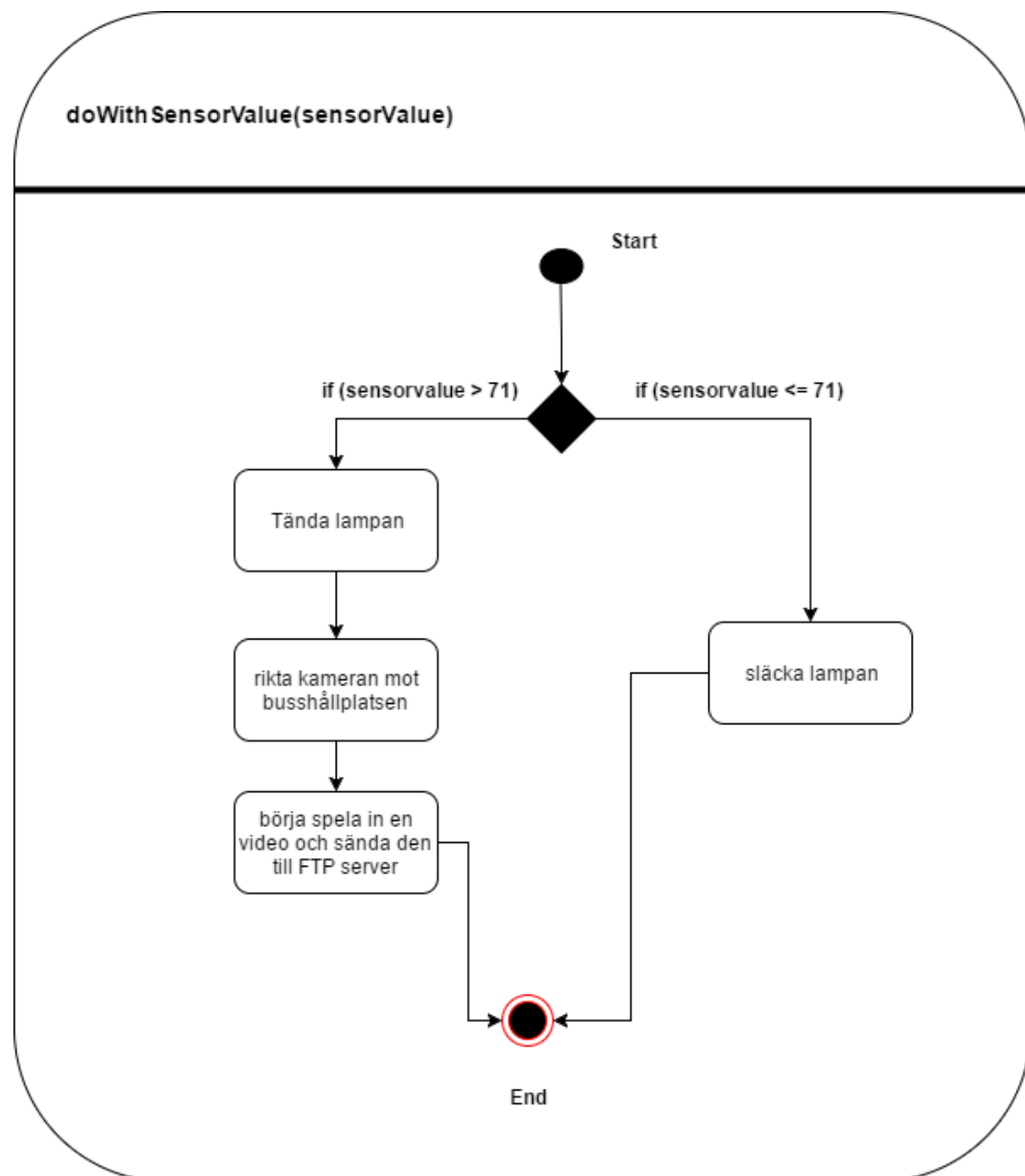
5.1 GitHub-länk

- <http://github.com/MalmoUniversity-DA264A/SAFE24.git/>

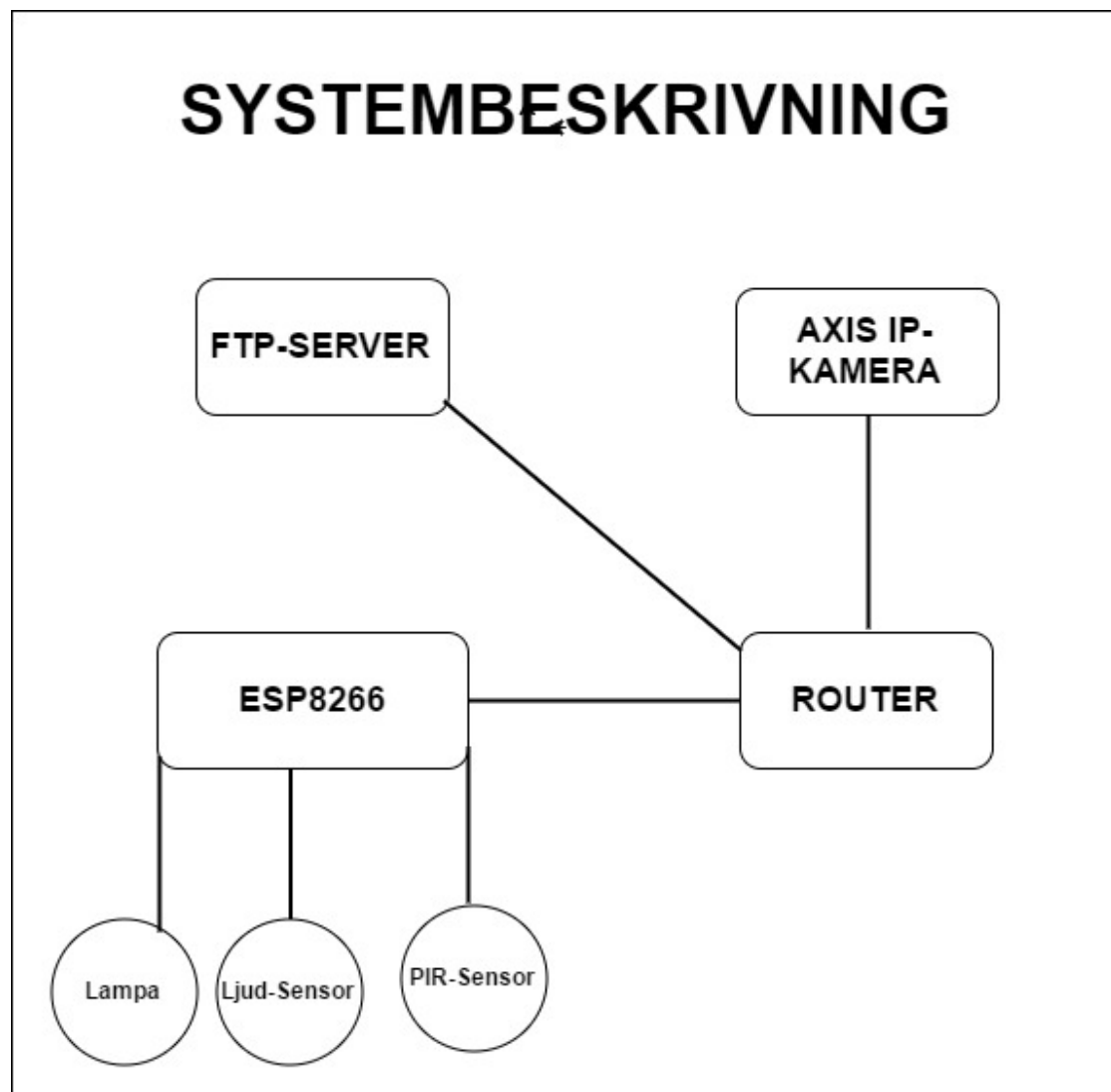
5.2 Diagram



Figur 5.1: Flödesdiagram för systemet.



Figur 5.2: Flödesdiagram för metoden `doWithSensorValues`.



Figur 5.3: Systembeskrivning

5.3 Testfall

ID	Title	Preconditions	Steps	Expected Result	Status
T9	connectWifi	Värdens SSID och nyckel behöver specificeras.	Anropa funktionen connectWifi med rätt typ av argument. Första argumentet SSID och andra argumentet en nyckel.	Utskrift: IP adress : <den lokala IP-adressen>	Passed
T10	connectionTimeOut	Detta testas i samband med [C5]. Om [C5] misslyckas så kommer detta testfallet att lyckas.	Anropa funktionen connectWifi med rätt typ av argument men fel argument information. Detta medför att en anslutning inte kommer att kunna upprättas.	Utskrift: Connections timedout... Couldn't connect to host. IP adress : <den lokala IP-adressen>	Passed
T12	checkConnection (success)	[C5] har lyckats.	Anropa funktionen checkConnection().	Returnerar: 1	Passed
T11	checkConnection (fail)	[C6] har lyckats.	Anropa funktionen ceckConnection().	Returnerar: 0	Passed

Figur 5.4: Testfall för anslutning

ID	Title	Preconditions	Steps	Expected Result	Status
T51	Skicka video från kameran till servern	[C5] och [C9] har lyckats.	Virtual port number måste ha värdet 9 och den måste aktiveras	videoklippet ska sparas i ftp-servern	Passed
T52	Skicka video från kameran till servern då servern är offline	[C5] och [C9] ska vara uppfyllda och [C12] ska vara uppfyllt	aktivera virtuell port nummer 9	ingen videoklipp sparas i ftp-servern	Passed
T53	Skicka video från kameran till servern efter att ha stängt av och sedan på kameran	[C5] och [C9] är uppfyllda (FTP server igång, en anslutning mellan kameran och servern upprätthålls) [C12] är uppfyllt. portRecord är satt till 9	1. aktivera virtuell port nummer 9 genom att få ljudsensorn att gå över värdet 75 2. Stäng av kameran och kort därefter slå på den igen 3. Skicka video till server	Video ska finnas i ftp-servern	Passed
T54	Skicka video från kameran till servern efter att servern har startas om	[C5] & [C9] är uppfyllda (FTP server igång) samt att [C12] är uppfyllt portRecord är satt till 9	1. Anslut till ftp-servern 2. aktivera virtuell port nummer 9 via ljudsensorn 3. Starta om servern och utför steg 1-2 om	2 videofiler kommer sparas i ftp-servern	Passed
T57	Kontrollera om videoklippets namn är formaterat på rätt sätt	[C5] och [C9] är uppfyllda (FTP server igång) och [C12] är uppfyllt portRecord är satt till 9	1. Skicka videoklipp till servern genom att anropa aktivera virtuell port nummer 9 via ljudsensorn 2. Kolla i ftp-servern om videoklippet har sparats på rätt sätt	videonamnet ska se ut så här: yy-mm-dd-hh:mm:ss Formatet ska vara .mkv	Passed
T58	Kontrollera videolängden om ljudsensorn aktiveras en gång	[C5] & [C9] är uppfyllda samt att [C12] är uppfyllt	1. aktivera virtuell port nummer 9	videoklipp ska sparas i ftp servern varslängd är 35 sekunder	Passed
T59	Kontrollera videolängden om ljudsensorn aktiveras två gånger inom 60 sekunder	[C5] & [C9] är uppfyllda samt att [C12] är uppfyllt	1. aktivera virtuell port nummer 9 2. aktivera virtuell port nummer 9 igen efter 30 sekunder	en videoklipp ska sparas med längden 1:30 min	Passed

Figur 5.5: Testfall för FTP server

ID	Title	Preconditions	Steps	Expected Result	Status
T75	sendToCamera (success)	[C5] har lyckats.	Anropa funktionen sendToCamera med rätt argument.	Utskrift: Connecting to <host> [HTTP] GET... code: <httpCode> <http_response> Returnerar: 1	Passed
T76	sendToCamera (fail)	[C5] lyckats.	Anropa funktionen sendToCamera med rätt argument förutom det tredje argumentet. Tredje argumentet måste vara felaktigt Anropa funktionen sendToCamera med rätt argument förutom det fjärde argumentet. Fjärde argumentet måste vara felaktigt	Kameran ska inte reagera på kommandot/http request	Passed
T77	activateVirtualPort	Testfall C5 och C10 lyckats.	Anropa metoden activateVirtualPort med rätt argument (portNumber = 9).	Utskrift: "/axis- cgi/virtualinput/act ivate.cgi?schemave rsion=1&port=" + portNumber	Passed
T78	deactivateVirtualPort	Testfall C5 och C10 lyckats.	Anropa funktionen deactivateVirtualPort med integer argumentet portNumber = 9.	Utskrift: /axis- cgi/virtualinput/de activate.cgi?schem aversion=1&port=" + <portNumber>	Passed
T93	Testar startriktningen	Koppla in kameran till en strömkälla	Koppla in kameran till en strömkälla	Riktad mot homepositionen	Passed

T94	Testar riktningen mot busshållsplatsen	[C5] och [C9] är uppfyllda portStation är satt till 8	få ljudsensorn att gå över värdet 71	Kameran riktas mot busshållsplatsen	Passed
T95	Testar om kameran är riktad mot busshållsplatsen i 5 sekunder	[C5] och [C9] är uppfyllda portStationOne är satt till 8	få ljudsensorn att gå över värdet 71	Kameran ska riktas mot busshållsplatsen i 5 sekunder	Passed
T96	Testar om kamerans route fungerar	[C5] och [C9] är uppfyllda	1. aktivera virtuell port nummer 9 2. aktivera virtuell port nummer 8 3. Vänta i 5 sekunder	ska röra sig horisontellt åt höger	Passed
T97	Testar om kameran riktning nollställs mot busshållsplatsens riktning	[C5] och [C9] är uppfyllda samt att portStationOne är satt till 8	1. få ljudsensorn att gå över värdet 71 2. Vänta i cirka 32 sekunder (då kommer virtuell port nummer 10 aktiveras och ställa om riktningen till busshållsplatsen riktning)	Kameran riktas mot busshållsplatsen	Passed
T98	Testar kamerans slutriktning	[C5] och [C9] är uppfyllda samt att portHome är satt till 10 i main.ino filen	1. få ljudsensorn att gå över värdet 71 2. Vänta i 35 sekunder (efter 35 sekunder kommer virtuell port nummer 10 aktiveras)	Kameran är riktad mot taket	Passed

Figur 5.6: Testfall för IP kamera

ID	Title	Preconditions	Steps	Expected Result	Status
T110	doWhenMove (rörelse detekterad)	Definiera rätt INPUT och OUTPUT pinnar på ESP8266-modulen samt anslut kablarna till rätt pins på ESP8266-modulen.	Anropa funktionen doWhenMove.	Returnerar: 1	Passed
T118	doWhenMove (ingen rörelse detekterad)	[C14] lyckats.	1. Anropa funktionen doWhenMove. 2. Omgivningen får inte röra sig kring PIR-sensorn.	Lampan slocknar Returnerar: 0	Passed
T111	doWithPirValue (rörelse detekterad)	[C14] lyckats.	Anropa doWithPirValue	Returnerar: 1	Passed
T122	doWithPirValue (ingen rörelse detekterad)	[C14] eller [C24] lyckats.	Anropa doWithPirValue.	Returnerar: 0	Passed
T112	doWithSensorValue (ljud detekterat)	Definiera rätt INPUT och OUTPUT pinnar på ESP8266-modulen samt anslut kablarna till rätt pins på ESP8266-modulen.	1. Anropa metoden doWithSensorValue med rätt argument från analogRead.	Returnerar: 1	Passed
T123	doWithSensorValue (inget ljud detekterat)	[C16] har lyckats.	1. Anropa metoden doWithSensorValue med rätt argument från analogRead.	Returnerar: 0	Passed

Figur 5.7: Testfall för sensorerna

ID	Title	Preconditions	Steps	Expected Result	Status
T148	Kontrollera om serial monitor ger utskrift från samtliga tasks då de körs parallellt	main.ino överförd till ESP8266	Starta serial monitor	Sensornvärden skrivs ut	Passed

Figur 5.8: Testfall för multitasking