

**Tillämpning av kamerasystem för detektering utav
skadegörelse vid utsatta busshållplatser och
trygghetsskapande vid behovs övervakning**



MALMÖ HÖGSKOLA

Yurdaer Dalkic
George Albert Florea
Louay Khalil
Benjamin Sejdic

19 February, 2017 Malmö

Abstract

Mot slutet på kursen Inbyggda System och Signaler fick studenter på andra läsåret ett examinationsarbete i samarbete med Axis i Lund. En avancerad IP-kamera, samt ett litet elektriskt kit med en mikroprocessor och några sensorer användes till detta projektet. Projektet inleddes med en föreläsning från två Axis-anställda som klargjorde att examinationsuppgiften var att komma på ett problem och en lösning till det problemet. Problemet som ska lösas är den otryggheten människor känner och vandaliseringen som finns omkring busshållplatser. Lösningen byggde främst på de komponenter som gruppen fick, och det fanns det utrymme för att köpa in mer komponenter. Gruppen heter SAFE24 bestående av Louay, Yurdaer, George och Benjamin.



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

Hittills har det inte varit några examinationsprojekt på Datateknik-ingenjörsutbildningen, med sådana komponenter som nämndes tidigare. Malmö Höskola har satsat mer på externa samarbeten, vilket gynnar skola, studenter och även de externa samarbetsparterna.

Grupp SAFE24 vill tacka de ansvariga för programmet som ser till att sådant kan gå i verk och vill även tacka Axis från Lund som stödjer elever och skolverksamhet rent generellt.

Innehåll

| | | |
|----------|--------------------------------------|-----------|
| 1 | Inledning | 1 |
| 2 | Teori | 2 |
| 2.1 | Problemet med skadegörelse | 2 |
| 3 | Material & Metoder | 4 |
| 3.1 | Metoder | 4 |
| 3.2 | ESP8266 - mikroprocessorn | 6 |
| 3.3 | FTP-Server | 8 |
| 3.4 | IP-kameran | 8 |
| 3.5 | Sensorer | 9 |
| 3.6 | Systembeskrivning | 12 |
| 3.7 | Arbetsuppgifter | 12 |
| 4 | Resultat | 14 |
| 4.1 | PIR-sensorn | 14 |
| 4.2 | Mikrofon | 14 |
| 4.3 | Testfall | 14 |
| 5 | Diskussion & Framtid | 15 |
| 5.1 | Diskussion | 15 |
| 5.1.1 | Lagar och Etsika Aspekter | 15 |
| | Litteraturförteckning | 17 |

| | | |
|----------|-----------------------|-----------|
| 6 | Bilaga | 18 |
| 6.1 | GitHub-länk | 18 |
| 6.2 | Diagram | 18 |
| 6.3 | Testfall | 22 |

1

Inledning

Många hållplatser i Malmö och andra städer är utsatta för vandalisering. Det handlar mest om hållplatser som är utanför stadens centrum eller utanför övervakningsområden. Skadegörelse kostar pengar eftersom vandaliseringen kräver reparationer. Gruppen (SAFE24) ska med hjälp av Axis IP-kamera lösa problemet med vandalisering och även otrygghet omkring busshållsplatserna med ett smart övervakningssystem.



Figur 1.1: En vandaliserad hållplats i Borås. (www.bt.se/)

2

Teori

2.1 Problemet med skadegörelse

Sannolikheten för vandalisering av busshållplatser ökar när det är mörkt ute och under nattetid just på grund av att brottet inte skall upptäckas. Det har visat sig att i ett område där det finns en skylt som säger att området är övervakat har brottsligheten minskat och människors uppförande förbättrats.

Brottsförebyggande rådet (BRÅ) skriver på sin hemsida att kameraövervakning i brottsförebyggande syfte blir allt mer vanligt i Sverige. Speciellt när kameraövervakningslagen trädde i kraft i juli år 2013. Lagändringen har underlättat för installationer av kameraövervakningssystem.



Figur 2.1: En obevakad hållplats i Malmös stadsdel Lindängen. (Google Maps: Lindängen - Malmö)

2.1 Problemet med skadegörelse

Busshållplatser i utsatta områden utan övervakning där människor passerar utsätts för skadegörelse och denna skadegörelse kostar samhället pengar. Skadegörelsen fortsätter än idag eftersom det inte finns några bra lösningar för att hantera problemet med skadegörelse. Detsamma gäller tryggheten omkring de busshållplatser som är utsatta för vandalisering.

En busshållplats som är övervakad dygnet runt är en ineffektiv lösning. Övervakningen ska ske i samband med att specifika villkor är uppfyllda. Villkoren för aktiv övervakning är då en människa är närvarande, rörelse registreras eller vandalisering mot busshållplatsen utförs. Om inget villkor är uppfyllt så kommer systemet att vara i ett passivt tillstånd och enbart lyssna på förändringar.

Sensorer används för att lyssna till förändringar hos omgivningen. Dessa förändringar kommer att utvärderas och jämföras med fördefinierade villkor för när övervakningssystemet ska aktiveras. När ett villkor är uppfyllt så kommer systemet att aktiveras och börja registrera data och skicka denna data via internet till en server för datalagring.

3

Material & Metoder

3.1 Metoder

För att öka tryggheten för väntande bussresenärer utvecklades ett system för att detektera närvaro.

En vibrationssensor användes först men den var för känslig så vibrationssensorn ersattes med en PIR-sensor istället. När värdet från PIR-sensorn överskred ett tröskelvärde så tändes en lampa. Lampan var tänd en viss förutbestämd tid innan den släcktes. Denna systemlösningen kommer att användas när aktiv övervakning ska ske.

För att upptäcka vandalisering av busshållplatsen användes en ljud-sensor.

IP-kameran var installerad i mitten av vägen för att först filma hållplatsen och sen omgivningen runt hållplatsen. Detta är viktigt eftersom vandaliseringen kan ske på avstånd.

En schemaläggare användes för att hela systemet skulle fungera samtidigt så att komponenter inte begränsade varandra. Varje komponent turades om en begränsad tid att vara aktiv. Detta var nödvändigt att göra eftersom sensorer skulle lyssna kontinuerligt på förändringar hos omgivningen medan systemet var aktivt och utförde andra uppgifter.

3.2 ESP8266 - mikroprocessorn

Huvudanledningen för valet av denna mikroprocessor var att den hade en inbyggd Wifi-mottagare vilket var absolut nödvändigt för att kunna kommunicera med nätverket där kameran var uppkopplad. Givetvis kunde gruppen hitta på alternativa lösningar men just denna lösning var den smidigaste. Det fanns tillräckligt med både digitala ingångar och en analog ingång. Den analoga ingången klarar en spänning på 1V.

Sensorer kopplades till ESP:n och ESP:n kommunicerade med kameran via wifi-anslutning.

Svårigheten som uppstod var hanteringen av avlästa sensor värden med avseende på tid och periodicitet. Koden nedan visar hur gruppen fick sina värden:

```
const int pirSen = 13; // Digital pin for pir sensor
...
pinMode(pirSen, INPUT); // The pin is set as INPUT
...
int pirValue = digitalRead(pirSen); // Gets a reading from the pin
```

Efter flera tester och noteringar kom gruppen fram till ett counter-system för att hantera sensorns digitila värden.

```
void doWithPirValue(int pirvalue) {
  if (pirvalue == HIGH){
    pirCounter = pirCounter + 1; // Counter for each reading
    pirSum = pirSum + 10; // Sum-counter for each HIGH value reading
  }

  if (pirvalue == LOW){
    pirCounter = pirCounter + 1; // Counter for each reading
  }
}

void doWhenMove() {
  if (pirCounter == 7) {
    pirCounter = 0; // Reset for the counter of each reading
    if (pirSum == 70) {
      prevTime = millis(); // Timer to put light on
    }
  }
}
```

```
    digitalWrite(pirLed, HIGH); // Big light ON
  }
  if (pirSum < 70 && ( millis() - prevTime ) > 10000 ) {
    digitalWrite(pirLed, LOW); // Big light OFF
    prevTime = 0; // Reset for the timer
  }
  pirSum = 0; // Reset for the sum-counter of HIGH value readings
}
}
```

ESP:n läste av ljud-sensorn (mikrofonen) analogt. Det uppstod under demo-dagen ett problem där mikrofonen hela tiden gav höga analoga värden. Det berodde på att sensorn hängde upp och ner och vid montering under redovisningen hade sladden från mikrofonen till ESP:n lossnat vilket medförde att ESP:n läste in analoga värden på 1023. Det löstes genom att filtrera bort analoga värdet på 1023 i koden med en if-sats och kopplade ESP:ns analoga pin till jord via en resistor. Resistorn hade en hög resistans så den inte påverkade sensorns avläsning. Koden nedan visar hur sensorn läser in sina värden.

```
const int micSen = A0; // Analogpin for the pir sensor
const int micLed = 12; // Small lamp for mic-action

int sensorValue = analogRead(micSen); // Gets an analog reading

void doWithSensorValue(int sensorvalue) {

  if ( sensorvalue < 1023 ) { // To make sure of correct reading
    if (sensorvalue > 71) { // limit value to alert
      digitalWrite(micLed, HIGH); // MIC light ON
    }
    if (sensorvalue < 72) { // limit value to non-alert
      digitalWrite(micLed, LOW); // MIC light OFF
    }
  }
}

}
```

3.3 FTP-Server

För att kunna identifiera personer begår vandalisering måste systemet lagra bilder/-inspelningar på en server. En FTP-server användes. FTP-servern och IP-kameran låg under samma subnät.

Information om FTP-servern:

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

3.4 IP-kameran

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.

IP-kameran användes för att skicka bilder och video till en server för datalagring.

Kommunikationen med kameran gjordes via ESP8266 som sände kommandon över internet för att styra kameran.

Tre events skapades; "ActionPTZStation1", "ActionRecord", "ActionPTZHome".

ActionPTZStation1: När virtuell port X aktiveras så riktas kameran till en bestämd position som heter plats1" (busshållplatsen).

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

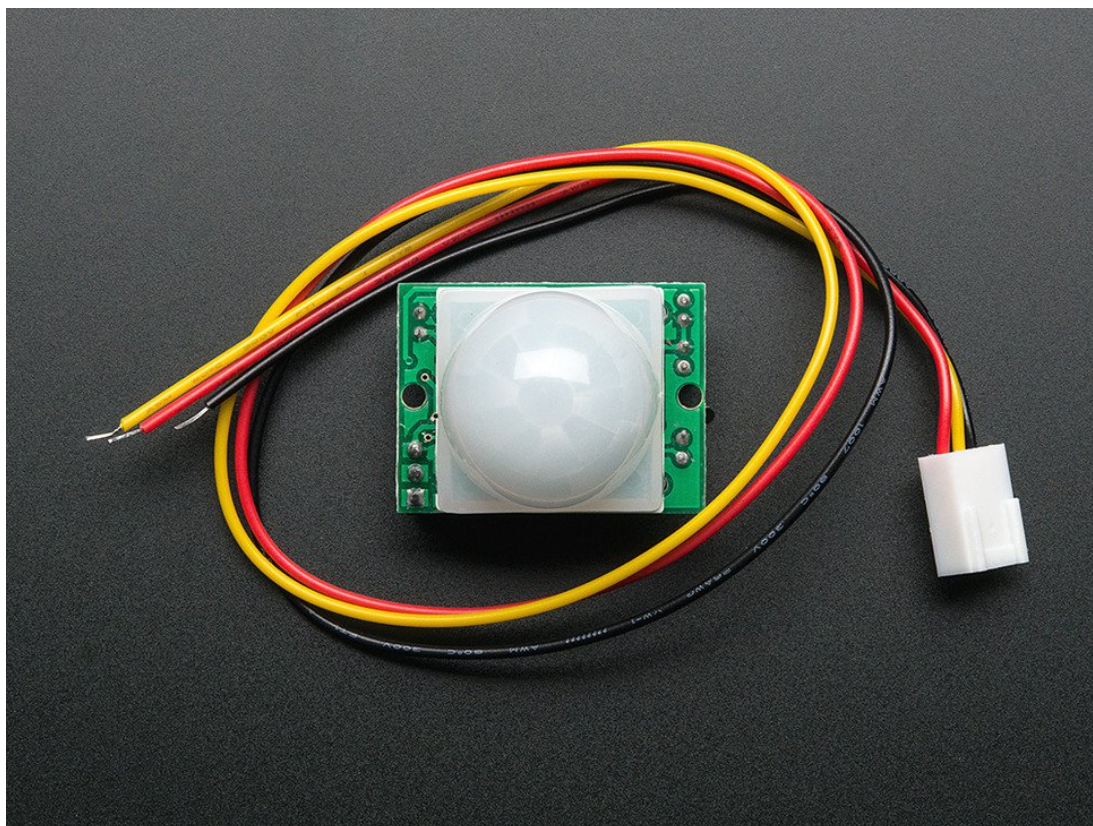
ActionPTZHome: När virtuell port X aktiveras så riktas kameran till en bestämd position som heter Safe24öch motsvarar start och slutpositionen för kameran.

3.5 Sensorer

En PIR-sensor från Adafruit användes som rörelsedetektor för att lysa upp den stora lampan. Den kopplades till ESP:n via tre sladdar, en röd som gick till strömmen, trots att det krävdes minst 5 V så gick det bra med 3,3 V från ESP:n. Den andra sladden gick till jord och den tredje är själva signalsladden som då gick till en digital pin som var satt till input-mode. För att gruppen skulle lyckas med pir-sensorn fick de med hjälp av delay och en counter räkna in hur många gånger som signalen är hög eller låg. Efter flera tester och försök kom de fram till att det behövdes minst 7 rundor där en runda är på 0,5 s. Anledningen till detta var att när det inte fanns rörelse så gav PIR-sensor utslag med 6 ettor och en nolla. Då det fanns rörelse gav den utslag på bara ettor. Slutligen gällde det att om programmet läste in 7 värden från signal-pin, så skulle summan vara 7 vid rörelse och om summan var under sju betydde det att ett av värden var noll, alltså ingen rörelse. Ibland kom det även störningar som gjorde att PIR-sensorn bara visade ettor, vilket senare fram i tiden löstes genom att utföra en digitalWrite på PIR-input-pin:n med värdet låg.

PIR-sensorn registrerade ifall det förekom någon rörelse. Denna sensorn skickade digitala värden till ESP8266.

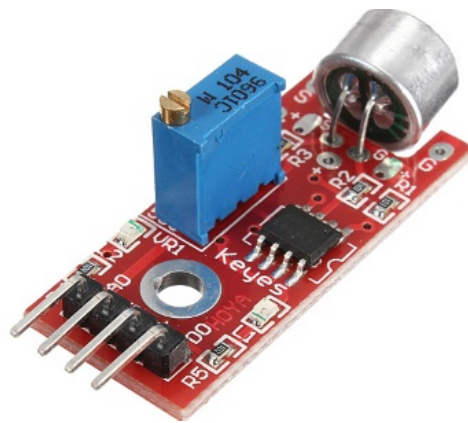
På grund av vibrationssensorn var så känslig använde gruppen en ljudsensor istället som hade mikrofon. På samma sätt som PIR-sensor kopplades den till ström och jord och en tredje sladd gick till analog-pin istället för en digital. Eftersom en digital pin gav antingen en låg eller hög signal, dvs antingen en etta eller en nolla, medan den analoga pin:n gav olika heltals-värden beroende på styrkan av signalen, som egentligen motsvarade för högre spänning. Glas har så som andra material en naturlig resonansfrekvens med vilken material oscillerar eller enklare sagt vibrerar. Självklart varierar denna frekvens från typ till annan beroende av glasets form och om det innehåller något dämpande. Ljud som har samma ton som den naturliga frekvensen kan få glas att börja



Figur 3.2: En PIR-sensor från Adafruit. (www.adafruit.com)

vibrera. Det krävs även en hel del ”styrka” vid sidan av tonen. Ju högre ljudet är desto mer vibrerar glaset, och vid en viss gräns kommer inte glaset att tåla dessa vibrationer och därför går glaset i sönder. Det krävs att man kommer upp till över 100 dB. Det säger egentligen inte mycket, om man inte vet i förväg att en människas normala tal är runtomkring 50 dB.

På grund av brist på glas och utrymme för gruppen för att de skulle göra försök att ta i sönder glas hade de bestämt i detta projekt att utgå från slag mot en kartong eller en miniatyr av en hållplats med en gräns på 75 dB. Över denna gräns betydde det att de har glas som hade gått i sönder.



Figur 3.3: En ljudsensor med mikrofon. (www.bazaargadgets.com)

3.6 Systembeskrivning

Nu har gruppen beskrivit olika delar av systemet och hur dem används. Om PIR-sensor detekterar någon rörelse då skall lampan ska tändas en viss tid och sedan släckas, så länge sensorn inte detekterar en ny rörelse. Ljud-sensorn är också igång hela tiden. Om ljudsensor detekterar ett ljud som är över gränsvärdet ska den viktiga processen börja. Först aktiveras två virtuella porter i kameran. Den ena riktar kameran mot busshållplatsen och den andra gör att kameran börjar filma och filmen skickas till FTP-servern. Videolängden är 40 sekunder och efter 5 sekunder börjar kameran att filma runt omkring dvs kameran ska börja vrida sig horisontellt. Ett varv tar cirka 27 sekunder dvs efter 33 sekunder ska kameran riktas mot busshållplatsen igen och det görs genom att aktivera en virtuell port. Kameran ska försätta filma cirka 7 sekunder till och efter det ska den riktas mot sitt standardläge dvs "homeposition". Ifall ljud sensor detekterar ett annat ljud som är över gränsen inom 40 sekunder då ska processen börja från början dvs kameran riktas tillbaka mot busshållplatsen, ska börja vrida efter 5 sekunder osv. Det enda som blir annorlunda vid detektering av två ljud inom 40 sekunder blir videolängden. Till exempel om ett annat ljud detekteras 15 sekunder efter första ljud då blir videolängden 40+15 sekunder dvs 55 sekunder.

Gruppen var väl medvetna om att ljudsensorn inte var det bästa valet för systemet, dock var den, den mest tillgängliga. Vid besök hos en elektrokit-grossist kunde de inte få tag på någon användbar trycksensor som var tillräckligt känslig för att ersätta ljudsensorn. Under demo-dagen fick de frågan om de hade tänkt på en Piezo-sensor, vilken är en typ av en tryck-sensor. Efter fakta-sökning fann gruppen att en Piezo-sensor hade varit ett bättre alternativ än ljudsensorn.

3.7 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

3.7 Arbetsuppgifter

av alla komponenter, förslag till router-lösning, byggandet av en hållplats och kontinuerlig testning av alla kopplingar.

4

Resultat

4.1 Testfall

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. Det vill säga att alla task kördes i rätt 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.

5

Diskussion & Framtid

5.1 Diskussion

5.1.1 Lagar och Etsika Aspekter

I Sverige det finns regler 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 huvudregeln är att tillstånd krävs om:

- kameran riktas mot ”en plats dit allmänheten har tillträde”
- utrustningen kan användas för personbevakning och
- kameran är uppsatt utan att manövreras 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 omfattats av lagen om allmän övervakning. Busshållplatser och gatorna räknas som allmänt platser enligt definitionen av allmänt plats. Det här ställer vissa krav på den som installerar och/eller äger det systemet som vi har skapat under den examinations projekt. Man måste se till att lagar och regler följs dvs man måste göra en ansökan om tillstånd till allmän kameraövervakning. Kameraövervakning är en metod som används för att minska brottsligheten. Effekterna varierar beroende på hur man arbetar med

kamerorna. Andra sidan är kameraövervakning alltid känsligt utifrån ett integritetsperspektiv. Det som är människor oroliga för när det gäller övervakningskameror är att integritet av människors privata liv. Det är människors privatliv som människor har rätt beskydda. En ingenjör bör respektera detta som alla andra människor. Ingenjörer har ansvar att verka för att tekniken används för samhällets och mänsklighetens bästa enligt hederskodexen för Sveriges ingenjörer [2]. Vi använder kameraövervakningen med ett syfte som är att bekämpa brott.

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>

6

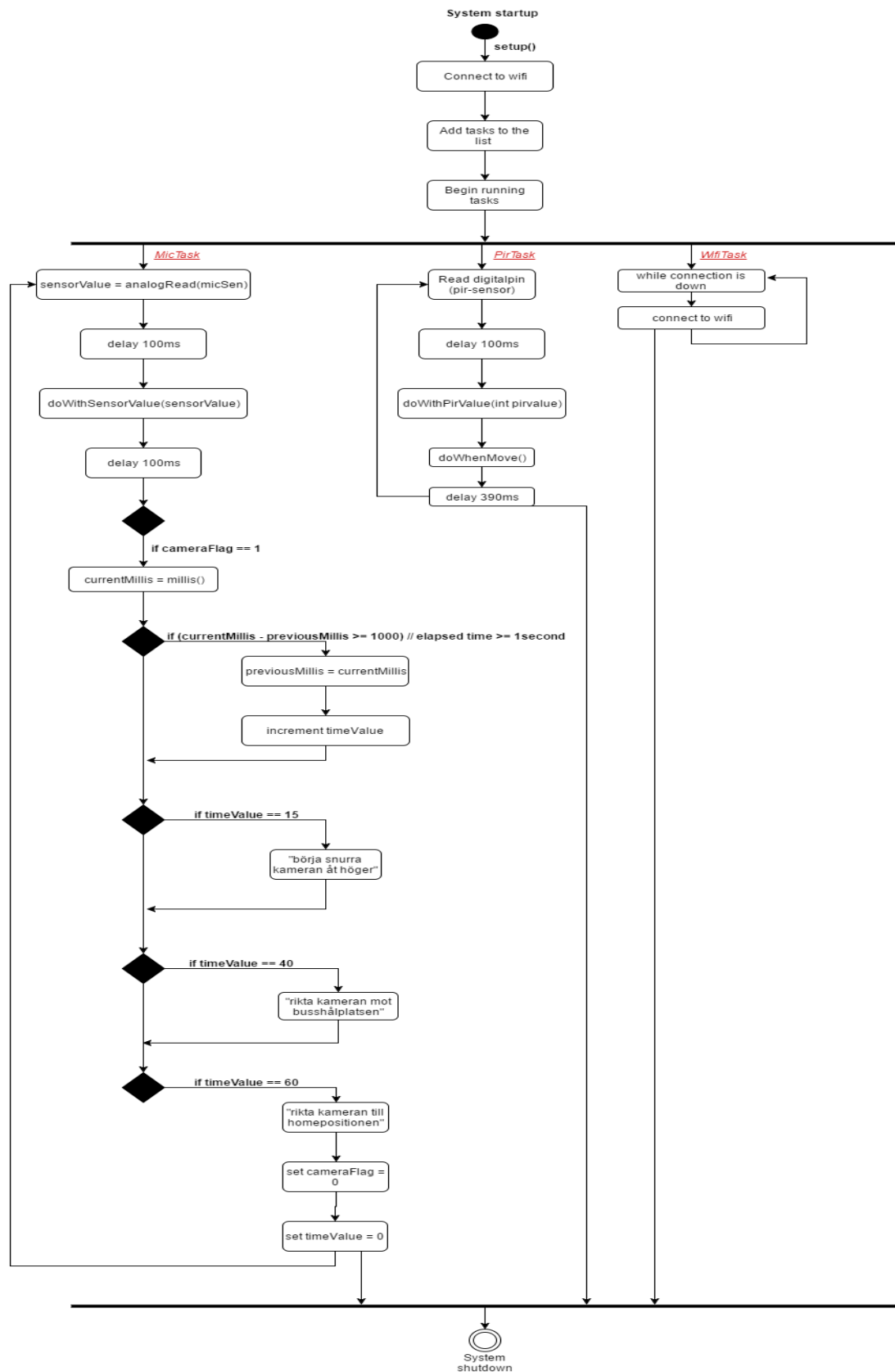
Bilaga

6.1 GitHub-länk

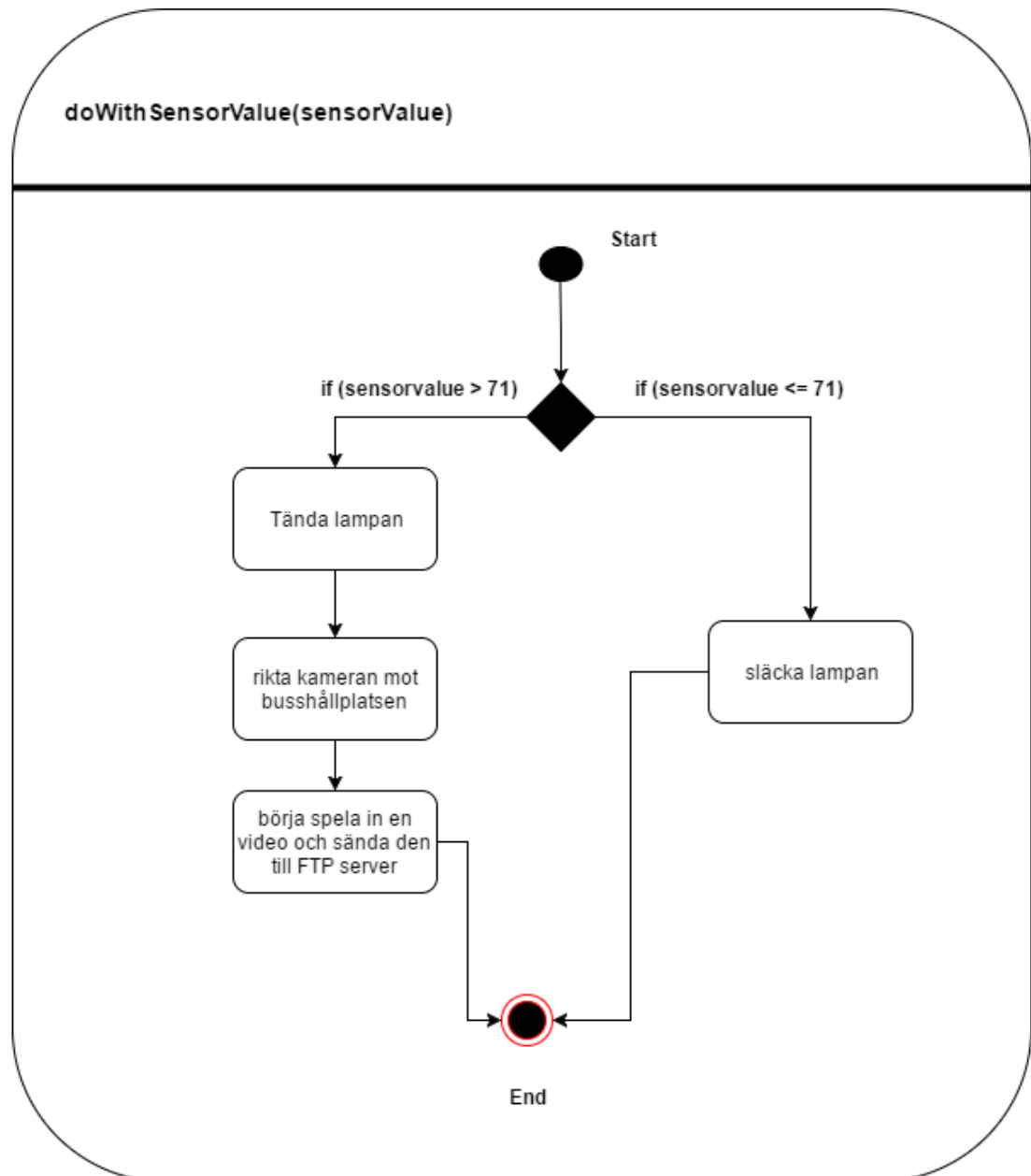
- <http://github.com/MalmoUniversity-DA264A/SAFE24.git/>

6.2 Diagram

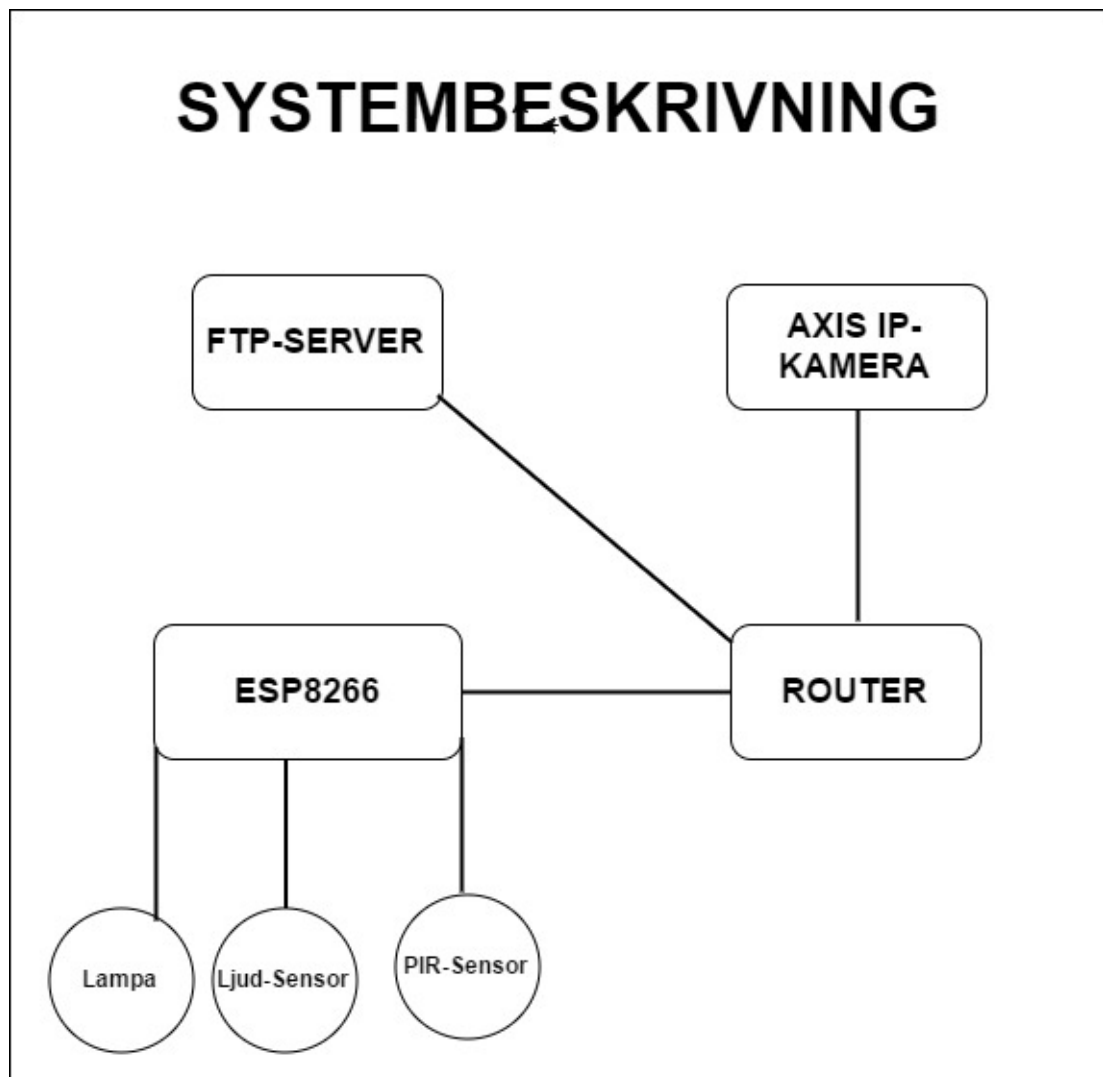
6.2 Diagram



Figur 6.1: Flödesdiagram för systemet.



Figur 6.2: Flödesdiagram för metoden doWithSensorValues.



Figur 6.3: Systembeskrivning

6.3 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 6.4: Testfall för anslutning

6.3 Testfall

| 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 | videoklipppet 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 videoklipppet 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 6.5: 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 homepostionen | 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 kamerans 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 6.6: Testfall för IP kamera

6.3 Testfall

| 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 6.7: 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 | Sensorvärden skrivs ut | Passed |

Figur 6.8: Testfall för multitasking