

**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 som läser andra året, ett examinationsarbete i samarbete med Axis i Lund. De fick låna en avancerad och en kostsam IP-kamera, samt ett litet elektriskt kit med en mikroprocessor och några sensorer. Projektet inleddes med en föreläsning från två Axis-anställda som sedan klargjorde att examinationsuppgiften var att komma på ett problem och en lösning till det problemet. En grupp, SAFE24, bestod av Louay, Yurdaer, George och Benjamin som sedan tidigare studier under utbildningens gång var bekanta med varandra. Det fanns olika idéer i gruppen om vad problemet skulle vara. Efter några möten kom det fram ett problem i samhället. Lösningen byggde främst på de komponenter som gruppen fick, dock fanns det utrymme för att köpa ytterligare fler förutsatt att det inte översteg en viss budget.



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	vi
2	Teori	vii
2.1	Problemet med skadegörelse	vii
3	Material & Metoder	ix
3.1	Metoder	ix
3.2	ESP8266 - mikroprocessorn	xi
3.3	FTP-Server	xiii
3.4	IP-kameran	xiii
3.5	Sensorer	xiv
3.6	Systembeskrivning	xvii
3.7	Arbetsuppgifter	xvii
4	Diskussion & Framtid	xix
4.1	Diskussion	xix
4.1.1	Lagar och Etsika Aspekter	xix
	Litteraturförteckning	xxi
5	Bilaga	xxii
5.1	GitHub-länk	xxii
5.2	Diagram	xxii
5.3	Testfall	xxii

1

Inledning

Många hållplatser i Malmö och andra städer är utsatta för vandalisering på olika möjliga sätt. Det handlar mest om hållplatser som är utanför stadens centrum eller utanför övervakningsområden. Det handlar om skadegörelser som kostar pengar, vilka kan utnyttjas till annat nödvändigt i staden. SAFE24, som kommer attt refereras till som gruppen härafter, sammanfattade sitt problem till att förhindra skadegörelser men även att hjälpa till att reda ut vem som ligger bakom en skadegörelse mot en busshållplats.



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 akten inte skall upptäckas. Det visar sig att brottsligheten minskar och bättre uppförande finns hos människor i ett område enbart om där finns en skylt som säger att området är övervakat. Gruppen bygger upp denna uppfattning eftersom Brottsförebyggande rådet BRÅ skriver på sin hemsida [3] att kameraövervakning i brottsförebyggande syfte blir allt mer vanligt i Sverige. Speciellt när kameraövervakningslagen trädde i kraft i juli 2013, har det underlättat för installationer av kameraövervakningar. Gruppen refererar även till egna erfarenheter om områden och tider på dygnet då människor känner sig otrygga vid busshållplatser.



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

Busshållplatser som är oövervakade där människor kommer och går i intervaller, kan utsättas för skadegörelse där exempelvis glaset krossas eller otillåten vandalisering görs. Denna skadegörelse kostar samhället pengar och fortsätter än idag att kosta samhället pengar då där inte finns någon bra lösning på problemet än. Tryggheten skulle möjligtvis öka då busshållplatsen är övervakad.

En busshållplats som är övervakad dygnet kommer att registrera mycket data och är således ineffektiv lösning. Kameran som kommer användas i syftet att övervaka busshållplatsen ska enbart aktivt övervaka busshållplatsen då specifika villkor är uppfyllda. Villkoren 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 något och jämföras med fördefinierade villkor för systemet. 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

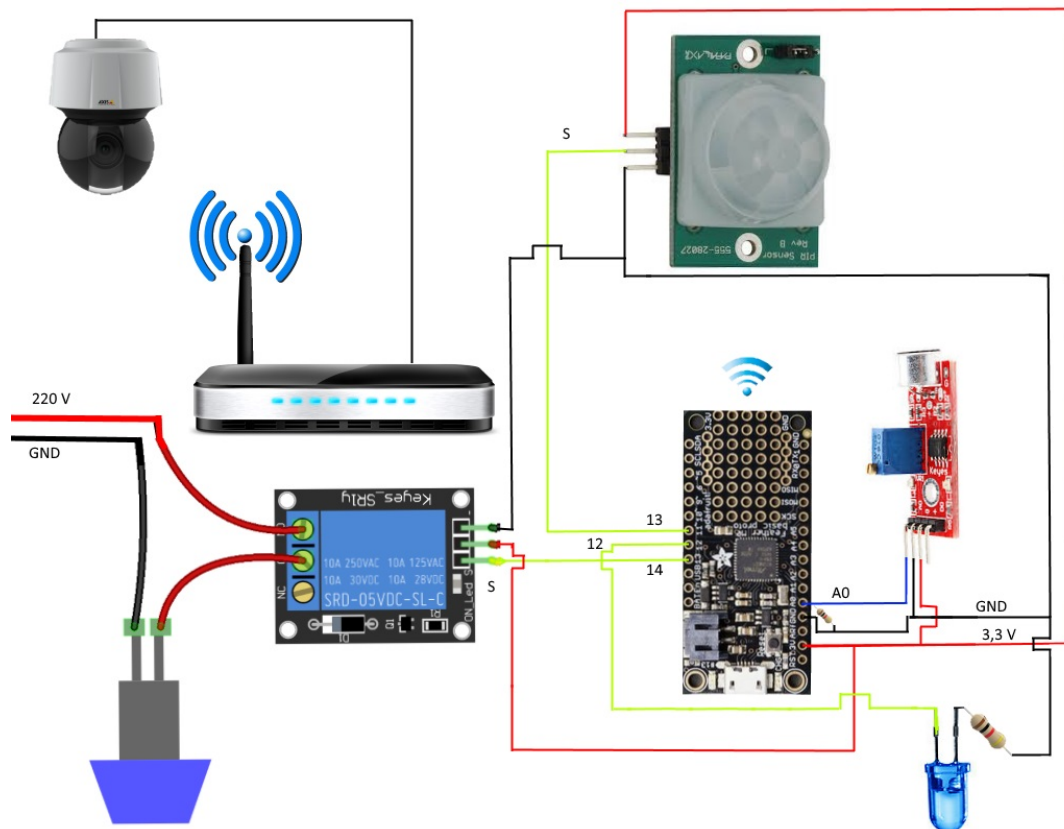
3.1 Metoder

Nu när gruppen hade ett problem återstod det att lösa problemet. I kittet fanns olika komponenter och sensorer. I början var uppmärksamheten riktad mot en vibrations-sensor för att upptäcka slag eller skadegörelse mot glasrutorna, dock visade det sig omedelbart att den var för känslig. Under tiden var det bestämt att använda en pir-sensor som skulle upptäcka rörelse inne i hållplatsen och därefter tända en lampa en viss period av tid. Tanken var att göra väntande busspassagerare trygga med hjälp av ljuset medan de väntade på bussen. Tillbaka till skadegörelsen av glasrutorna för att nu hade gruppen bestämt sig för att använda en ljud-sensor för att upptäcka slag eller skadegörelse mot en glasruta. En IP-kamera var installerad i mitten av vägen för att först filma hållplatsen och även runt omkring i ett varv om 360 grader. För att hela systemet skulle kunna fungera samtidigt så användes en schemaläggare så att varje komponent kunde vara aktiv utan att begränsa andra komponenter. Detta var nödvändigt att göra eftersom sensorer som användes skulle lyssna kontinuerligt på förändringar hos omgivningen medan systemet var aktivt och utförde andra uppgifter.

Pir-sensors uppgift var att lysa upp busshållplatsen vid rörelse i den. Ljud-sensors uppgift var att tala om för mikroprocessorn att aktivera IP-kameran så den filmade och skickade iväg filmen till en server. Därför upprättades en FTP-server för att kunna ta emot inspelningar och olika data från kameran och lagra dessa på en dator. Det behövdes ett nätverk för att åstadkomma kommunikation mellan de olika delar-

na. Mikroprocessorn med sina sensorer tillsammans med kameran och FTP-servern var uppkopplade inom samma nät, delvis via WIFI-anslutning och delvis med direkt-kabelanslutning.

De finns olika tasks i systemet som exekveras parallellt för att kunna låta processorn jobba med flera saker samtidigt. Det finns totalt 3 tasks i systemet varvid två av de (pirsensorn och ljudsensorn) agerar som inputs till systemet. Den tredje tasken (WifiTask) kontrollerar om det finns en wifi-anslutning, om anslutningen är nere försöker den återansluta till wifi. Biblioteket som möjliggör att dessa tasks arbetar oberoende av varandra och som detta systemet använde sig av skapades av Nicholas Wiersma och heter ESP8266Scheduler.



Figur 3.1: Alla komponenter kopplade med varandra.)

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. I övrigt fanns det allt de behövde till deras projekt. Det fanns gott om digitala pins samt en analog-pin, dock tog den emot max 1 V.

Deras tanke var att koppla sensorer till ESP:n som sedan kommunicerade med kameran utifrån de uppgifter som lästes in från sensorerna. ESP:n var då kopplad till samma nätverk som kameran och kommunicerade och skickade kommandon till kameran. Att läsa värden från sensors-pin:n var inget problem, dock fanns det problem med själva hanteringen av dessa värden rent tidsmässigt och periodiskt. 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
}
}
```

När det gällde mikrofonen så läste gruppen av sensorn analogt. Det uppstod under demo-dagen ett problem där mic-sensorn gav höga analoga värden konstant utan stopp. Det berodde på att sensorn hängde upp och ner och vid montering under redovisningen hade sladden till den analoga pin:n lossnat vilket medförde att ESP:n läste in analoga värden på 1023. Det löstes genom att först filtrera bort detta värde i koden med en if-sats och i kretsen genom att koppla ESP:ns analoga pin till jord via en resistor med en hög resistans så den inte påverkade sensorns avläsning när den var korrektkopplad. 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 som ligger bakom skadegörelser behövde systemet lagra bilder eller/och filmer på en server och den var en viktig del av lösningen. Gruppen valde att använda en FTP-server där bilder och/eller filmer skulle lagras. FTP-servern låg under samma subnät som IP-kameran. Information om FTP-servern:

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

3.4 IP-kameran

Kameran som används var tillhandahållen av AXIS. Kameran var en Q6128-E Network Camera med möjlighet till att via internetuppkoppling sända bilder och streama video till en server.

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.

Grupperna skapade tre aktiviteter i kameran, "ActionPTZStation1", "ActionRecord", "ActionPTZHome". ActionRecord var en aktivitet som spelade in en film som var en minut lång och skickade den till FTP-server som har namnet FTP-Safe24. Filmens upplösning som skickades till FTP-servern var 3840x2860. Ett suffix lades i filmen som innehöll datum och tidsinformation. ActionRecord aktiverades när virtuell port 9 var aktiv.

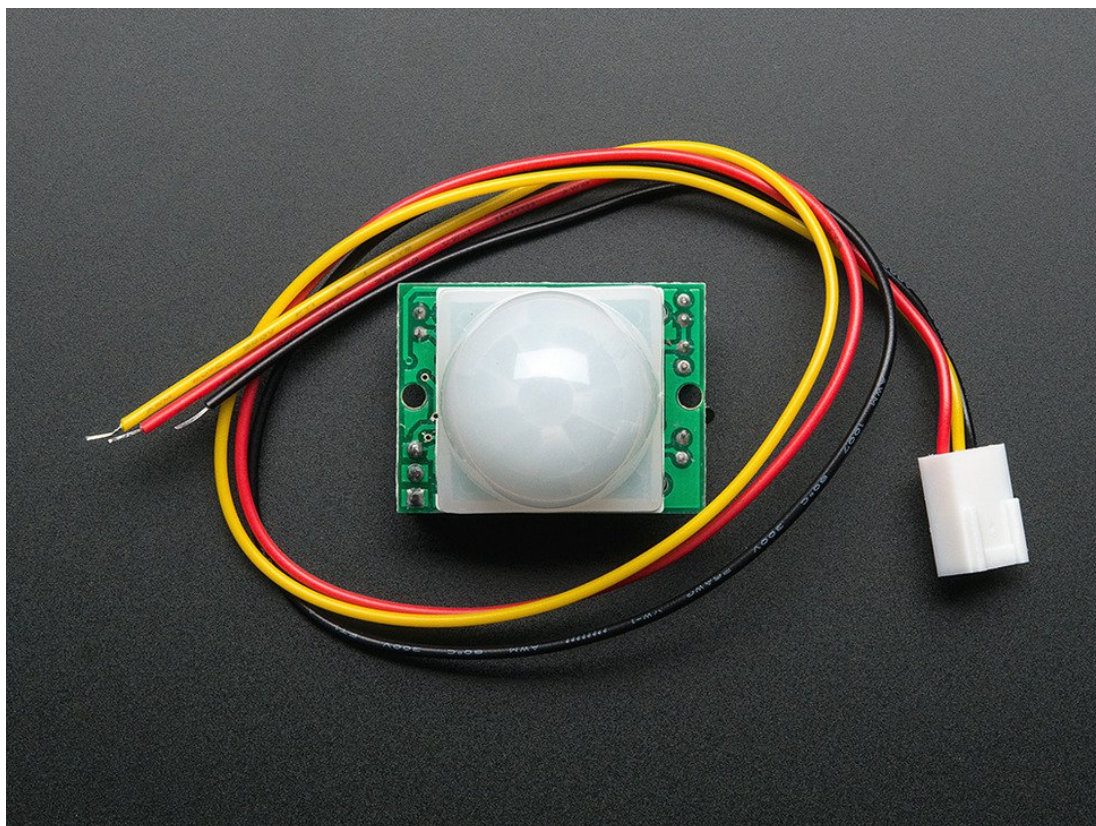
ActionPTZHome var den som riktade kameran till hemposition. Kamerans hemposition var definierad som position "Safe24". ActionPTZHome kunde aktiveras genom att aktivera virtuell port nummer 10. ActionPTZStation1 var den som riktade kameran till position som heter "plats1" (busshållplatsen). ". ActionPTZHome kunde aktiveras genom att aktivera virtuell port nummer 8.

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.

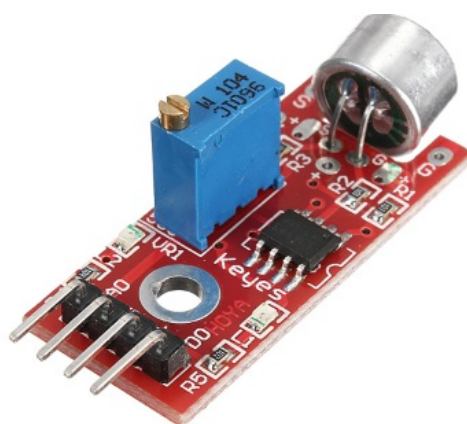
På grund av vibrationssensorn var så känslig använde gruppen en ljudsensor istället som hade mikrofon. På samma sätt som PIR-sensor kopplades den till ström och jord och en tredje sladd gick till analog-pin istället för en digital. Eftersom en digital pin gav antingen en låg eller hög signal, dvs antingen 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 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



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

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 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 annarlunda 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 välmedvetna 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

Diskussion & Framtid

4.1 Diskussion

4.1.1 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>

5

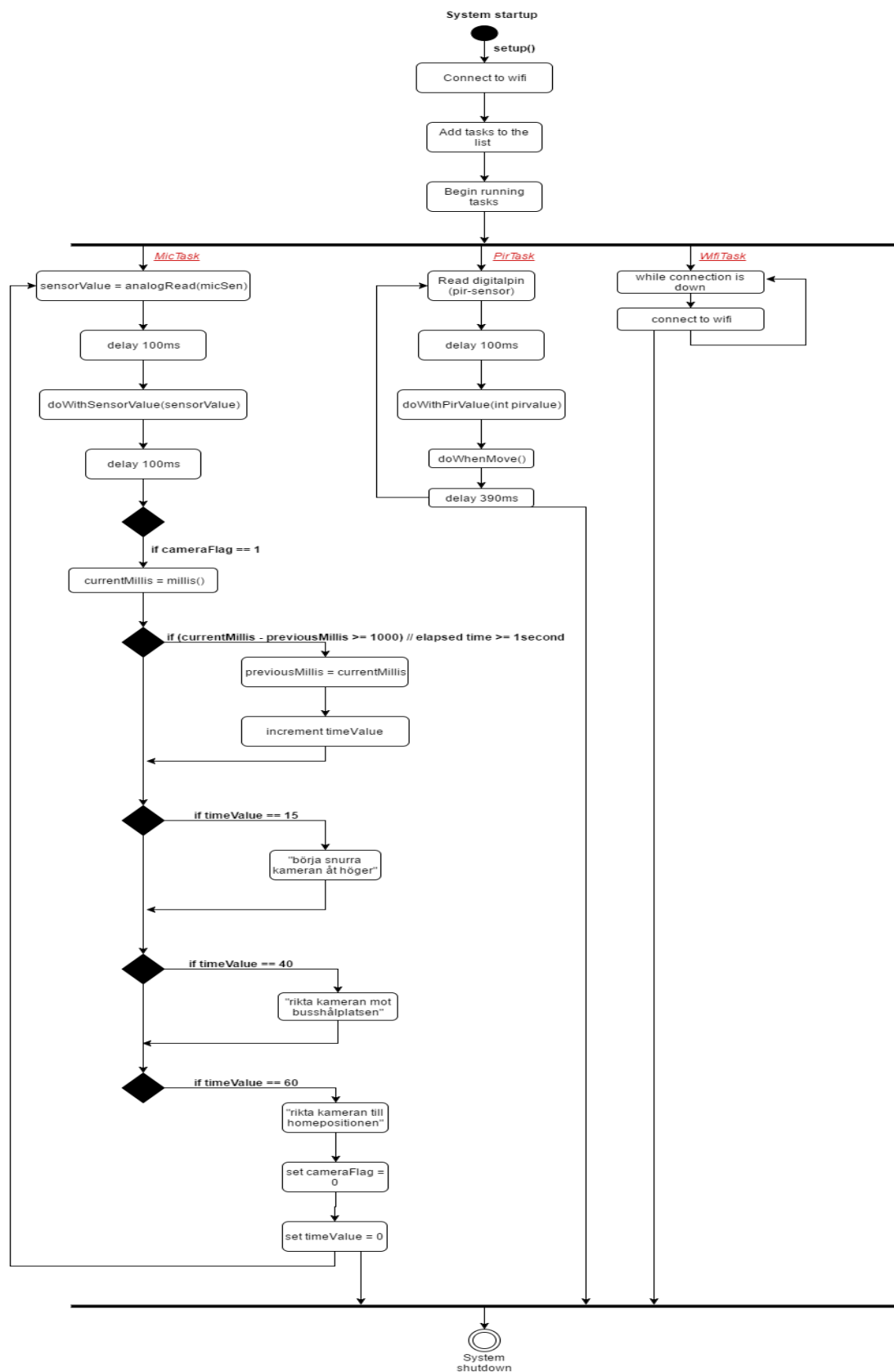
Bilaga

5.1 GitHub-länk

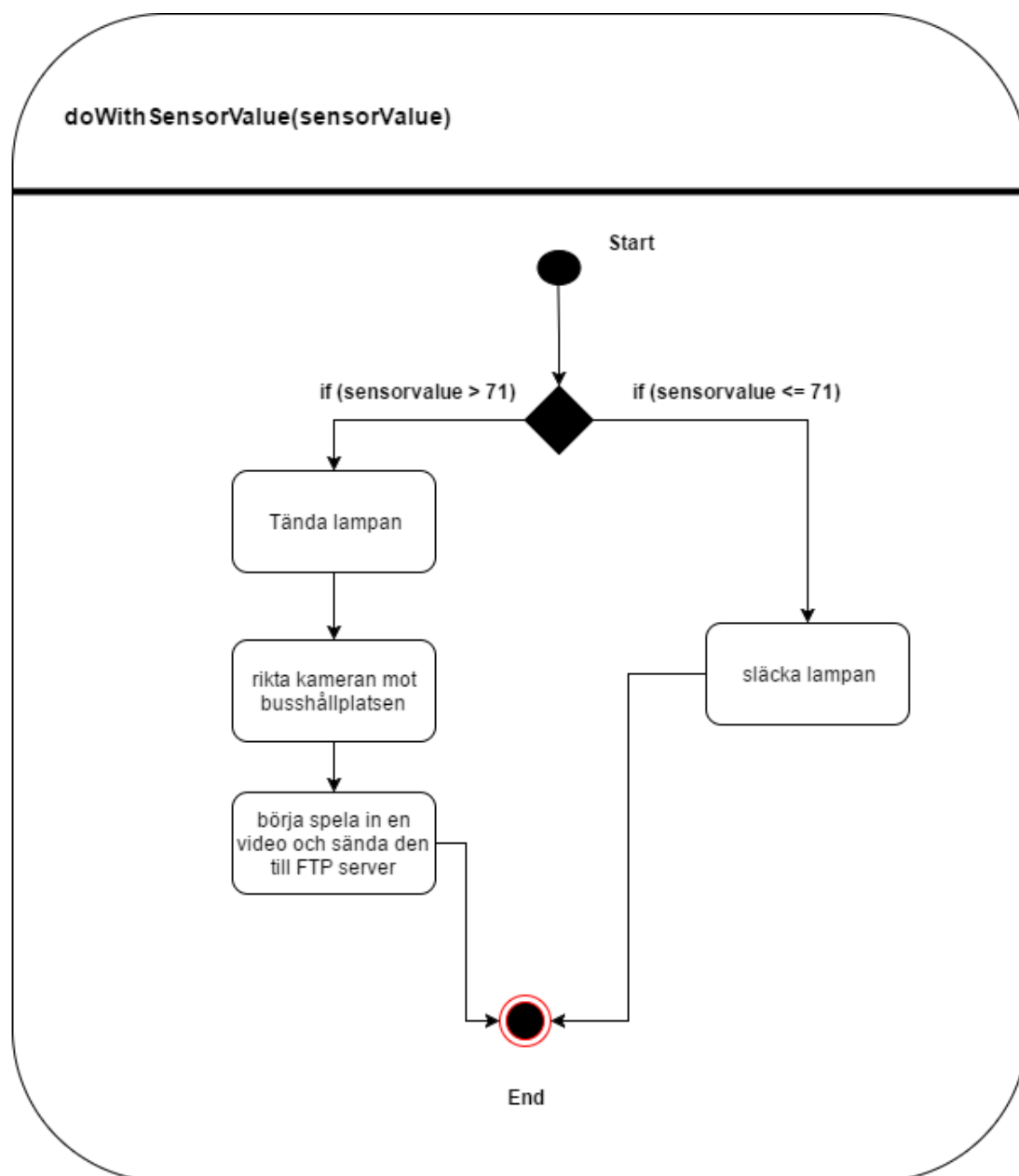
- <http://github.com/MalmoUniversity-DA264A/SAFE24.git/>

5.2 Diagram

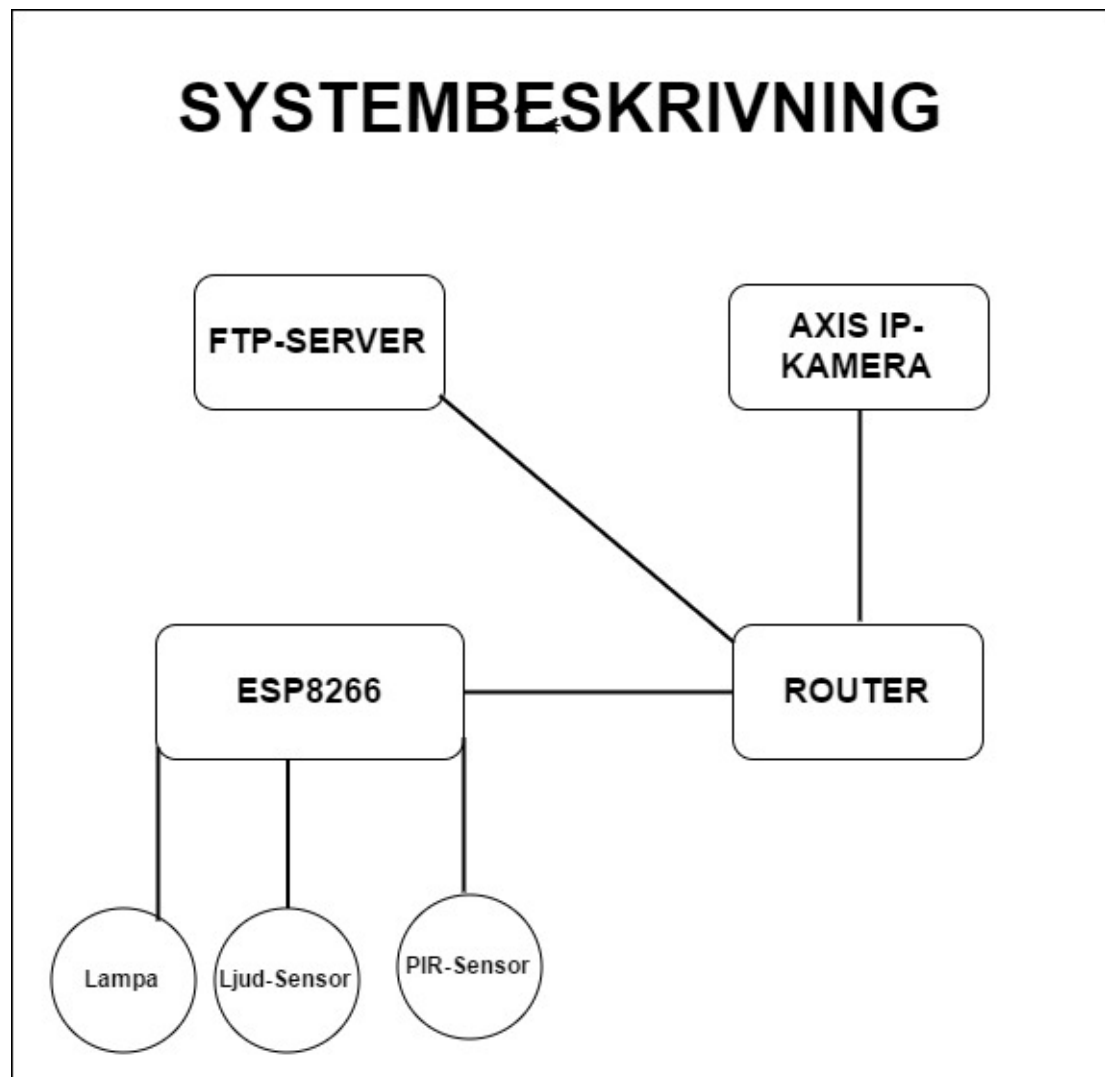
5.2 Diagram



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
C5	connectWifi	Vårdens 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>
C6	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.	Connections timedout... Couldn't connect to host. IP adress : <den lokala IP-adressen>
C9	checkConnection (success)	[C5] har lyckats.	Anropa funktionen checkConnection().	Returernar: 1
C7	checkConnection (fail)	[C5] har lyckats.	Anropa funktionen checkConnection().	Returernar: 0

Figur 5.4: Testflöde för anslutning

ID	Title	Preconditions	Steps	Expected Result
C19	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
C20	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
C21	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
C22	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
C25	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
C26	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 vars längd är 60 sekunder
C27	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

Figur 5.5: Testfall för FTP server.

5.3 Testfall

ID	Title	Preconditions	Steps	Expected Result
C10	sendToCamera (success)	[C5] har lyckats.	Anropa funktionen sendToCamera med rätt argument.	Utskrift: Connecting to <host> [HTTP] GET... code: <httpCode> <http_response> Returnerar: 1
C11	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	Utskrift: connection failed Returnerar: 0
C12	activateVirtualPort	Testfall C5 och C10 lyckats.	Anropa metoden activateVirtualPort med rätt argument (portNumber = 9).	Utskrift: "/axis-cgi/virtualinput/activate.cgi?schemaversion=1&port=" + portNumber
C13	deactivateVirtualPort	Testfall C5 och C10 lyckats.	Anropa funktionen deactivateVirtualPort med integer argumentet portNumber = 9.	Utskrift: /axis-cgi/virtualinput/deactivate.cgi?schemaversion=1&port=" + <portNumber>
C31	Testar startriktningen	Koppla in kameran till en strömkälla	Koppla in kameran till en strömkälla	Riktad mot homepositionen
C32	Testar riktningen mot busshållplatsen	[C5] och [C9] är uppfyllda portStation är satt till 8	få ljudsensorn att gå över värdet 75	Kameran riktas mot busshållplatsen
C33	Testar om kameran är riktad mot busshållplatsen i 15 sekunder	[C5] och [C9] är uppfyllda portStationOne är satt till 8	få ljudsensorn att gå över värdet 75	Kameran ska riktas mot busshållplatsen i 15 sekunder
C34	Testar om kamerans route fungerar	[C5] och [C9] är uppfyllda	1. aktivera virtuell port nummer 9 2. aktivera virtuell port nummer 8 2. vänta i 15 sekunder	ska röra sig horisontellt åt höger
C35	Testar om kameran riktning nollställs mot busshållplatsens riktning	[C5] och [C9] är uppfyllda samt att portStationOne är satt till 8	1. få ljudsensorn att gå över värdet 75 2. vänta i cirka 40 sekunder (då kommer virtuell port nummer 10 aktiveras och ställa om riktningen till busshållplatsen riktning)	Kamerans är riktad mot busshållplatsen
C36	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 75 2. vänta i 60 sekunder (efter 60 sekunder kommer virtuell port nummer 10 aktiveras)	Kameran är riktad mot taket

Figur 5.6: Testfall för IP kamera.

5.3 Testfall

ID	Title	Preconditions	Steps	Expected Result
C14	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
C24	doWhenMove (ingen rörelse detekterad)	[C14] lyckats.	1. Anropa funktionen doWhenMove. 2. Omgivningen får inte röra sig kring PIR-sensorn.	Lampan slocknar Returnerar: 0
C15	doWithPirValue (rörelse detekterad)	[C14] lyckats.	Anropa doWithPirValue	Returnerar: 1
C28	doWithPirValue (ingen rörelse detekterad)	[C14] eller [C24] lyckats.	Anropa doWithPirValue.	Returnerar: 0
C16	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
C29	doWithSensorValue (inget ljud detekterat)	[C16] har lyckats.	1. Anropa metoden doWithSensorValue med rätt argument från analogRead.	Returnerar: 0

Figur 5.7: Testfall för sensorerna.

ID	Title	Preconditions	Steps	Expected Result
C23	Kontrollera om serial monitor ger utskrift från samtliga tasks då de körs parallellt	main.ino överförd till ESP8266	Starta serial monitor	sensorvärden skrivs ut

Figur 5.8: Testfall för multitasking.