

# Återanvändbart HMI-program till staplingsrobot



---

**Philip Jonsson**

Division of Industrial Electrical Engineering and Automation  
Faculty of Engineering, Lund University





LUNDS  
UNIVERSITET  
Lunds Tekniska Högskola



## Examensarbete

# Återanvändbart HMI-program till staplingsrobot

Av  
Philip Jonsson

Avdelningen för Industriell Elektroteknik och Automation

Lunds Tekniska Högskola, Lunds universitet

SE-221 00 Lund, Sweden

© Copyright Philip Jonsson

LTH School of Engineering  
Lund University  
Box 882  
SE-251 08 Helsingborg  
Sweden

LTH Ingenjörshögskolan vid Campus Helsingborg  
Lunds universitet  
Box 882  
251 08 Helsingborg

Printed in Sweden  
Lunds universitet  
Lund 2022

# Sammanfattning

Uppgiften med examensarbetet är att skapa ett HMI (Human Machine Interface) till en robot som ska stapla lådor på en pall enligt angivet staplingsmönster. Funktionerna som var nödvändiga för att få HMI-programmet att fungera kommer att gås igenom. Även mycket av hur programmet är uppbyggt och hur de olika problemen löstes kommer tas upp. Arbetet undersöker även om programmet skulle kunna göras återanvändbart, så det bara är att skala upp eller ner programmet till nästkommande staplingsrobot.

Resultatet blev ett HMI-program där alla tänkbara parametrar går att ändra. Allt ifrån de olika lådornas längd och bredd till vilken ordning man ska ladda de olika staplingsmönstren på pallen går att bestämma. Programmet är även skalbart inom rimliga gränser, men kan skalas upp ytterligare med få modifikationer.

Det går även att utläsa en del statistik och beroende på ens roll i systemet får man tillgång till funktioner som kan användas vid till exempel felsökning.

Nyckelord: Staplingsrobot, HMI, PLC, iX Developer

# Abstract

The task of this Bachelors thesis was to create a HMI (Human Machine Interface) to a robot that shall stack boxes on a pallet after a given stacking pattern. The functions that was necessary to get the HMI-program to work will be gone through. A lot of how the program is built and how the different problems were solved will also be mentioned. The work will contain an investigation if it is possible to make the program reusable, so one only needs to scale it for the next stacking robot.

The result was an HMI-program where all imaginable parameters is changeable. Everything from the boxes length and width to which order the different stacking patterns should be stacked is changeable. The program is also scalable within reasonable limits but could be upscaled even more with small modifications.

It is also possible to read some statistics and, depending on one's role in the system, get access to functions that could be used for example troubleshooting.

Keywords: Stacking robot, HMI, PLC, iX Developer

# Innehållsförteckning

Sammanfattning.....	2
Abstract .....	3
Förord 6	
1. Inledning.....	7
1.1 Bakgrund.....	7
1.2 Syfte .....	7
1.3 Målformulering .....	8
1.4 Problemformulering .....	8
1.5 Avgränsningar .....	8
1.6 Motivering av examensarbetet .....	8
1.7 Resurser .....	9
2. Teknisk bakgrund.....	10
2.1 HMI .....	10
2.2 PLC.....	10
2.3 Mjukvaran .....	10
2.4 Hårdvaran .....	11
3. Metod .....	12
3.1 Förberedelser .....	12
3.2 Programmering.....	12
3.2.1 Taggar.....	12
3.2.2 Lådans dimensioner.....	13
3.2.3 Lådans position på pallen .....	13
3.2.4 Recept.....	13
3.2.5 Skärmar .....	13
3.3 Inhämtning av information.....	14
3.4 Källkritik .....	14
4. Analys.....	15
4.1 HMI .....	15
4.2 Problem .....	17
5. Resultat.....	19
5.1 HMI .....	19
5.1.1 Bakgrund .....	19
5.1.2 Visualisering.....	20
5.1.2 Översikt .....	25

5.1.3 Recept.....	26
5.1.4 Statistik.....	36
5.1.5 Säkerhet.....	38
5.1.6 Alarm.....	41
6. Slutsats .....	43
6.1 Reflektion över etiska aspekter .....	43
6.2 Framtida utvecklingsmöjligheter .....	43
7. Terminologi.....	45
8. Källförteckning.....	46
9 Appendix .....	47



# Förord

Mitt examensarbete har varit väldigt lärorikt och skulle vilja tack Prepac AB som lät mig genomföra det på deras företag.

Tack till alla anställda på Prepac som tagit väl hand om mig, men framför allt tack till Per och Erik. Utan deras hjälp och förtroende för mig så hade jag inte klarat av detta projektet.

Jag vill även tacka min handledare Bengt Simonson och min examinator Mats Lilja som hjälpt mig med mitt examensarbete och för deras undervisning under min studietid på LTH, Campus Helsingborg.

# 1. Inledning

I inledningen kommer bakgrunden, syfte, mål- och problemformulering samt avgränsningar för examensarbetet att beskrivas.

## 1.1 Bakgrund

Prepac AB är ett företag i Örkelljunga som bygger och levererar automationslösningar till sina kunder. Ibland gör de allt själva, hittar lösning, programmerar och konstruerar robotar samt installerar den åt kunden, men det kan även hända att de endast monterar och levererar en robot.

I deras nuvarande projekt ska en robot kunna plocka lådor och sedan stapla dessa på en pall efter ett angivet pallmönster. Examensarbetets huvudsakliga uppgift är att skapa ett HMI till denna robot men eftersom HMI-, PLC- och robotprogrammet ska kommunicera med varandra så krävs det en god förståelse för hur alla är uppbyggda. HMI står för Human Machine Interface och möjliggör kommunikation mellan människa och maskin med hjälp av en touch-skärm där ett grafiskt gränssnitt programmerats.

För att skapa HMI-programmet så kommer mjukvaruprogrammet iX Developer användas. Det är ett program framtaget av Beijer Electronics. Touch-skärmen där HMI-programmet ska synas är även den ifrån Beijer Electronics.

Företaget har en önskan att HMI-programmet ska göras återanvändningsbart till kommande projekt där det krävs stapling av lådor på en pall. Det ska finnas med i åtanke från början av arbetet.

All mjuk- och hårdvara kommer Prepac AB att tillhandahålla.

## 1.2 Syfte

Syftet med examensarbetet är att skapa ett HMI-program till en robot som ska stapla lådor på en pall efter angivet pallmönster på touch-skärmen. Programmet ska även vara återanvändningsbart till kommande staplingsrobotar.

## 1.3 Målformulering

Målet med examensarbetet är att skapa ett fungerande HMI till roboten som företaget ska leverera. Programmet ska i sin tur vara så pass återanvändningsbart så att tiden det tar att skapa ett HMI till framtida staplingsrobotar minskar.

## 1.4 Problemformulering

Under arbetets gång kommer följande frågor att besvaras:

1. Vilka funktioner och parametrar måste finnas med i HMI-programmet?
2. Hur ska programmet vara utformat för att minimera risker och fel vid körning av roboten?
3. Går det skapa ett skalbart program till pallerteringsapplikationer?
4. Hur ska visualiseringen ske för att man ska kunna skapa staplings- och pallmönster till roboten?

## 1.5 Avgränsningar

HMI-programmet är skapat med mjukvaran iX Developer och kommer därför bara fungera i framtiden om samma mjukvara används.

Man kommer maximalt kunna skapa 2 olika staplingsmönster med maximalt 12 lådor per lav och 16 lav per pall med hjälp av programmet. Skulle fler önskas i framtiden så kommer det krävas en liten modifikation av register och tillvägagångssätt.

För att stapla lådorna på pallen kommer roboten bara använda 4 ingångsvinklar (0, 90, 180 och 270 grader).

## 1.6 Motivering av examensarbetet

Automationsteknik och programmering har varit en stor del utav utbildningen och det är just vad detta arbete ska handla om. Det kommer innehålla utveckling av mjukvara som sedan ska testas på hårdvaran. Att få vara med och se hela arbetsprocessen, från första mötet till leverans, var en stor bidragande faktor till valet av arbete. Även företagets storlek och geografiska position spelade en roll när beslutet skulle tas.

## 1.7 Resurser

För att arbetet ska kunna genomföras krävs följande resurser:

- Mjukvara
- Hårdvara
- PC
- Lokal
- Handledare att kunna fråga om råd
- Utbildning av mjukvara och hårdvara

## 2. Teknisk bakgrund

I den tekniska bakgrunden kommer mjukvaran och hårdvaran som användes för att genomföra arbetet att gås igenom. HMI och PLC kommer också att förklaras i kapitlet.

### 2.1 HMI

HMI står för Human Machine Interface vilket skulle kunna översättas till Människa Maskin Gränssnitt på svenska. Det används för att skapa en koppling mellan människan och maskinen. Om det händer något så ska maskinen kunna tala om det för människan på ett sådant sätt att den förstår vad som hänt. Detsamma gäller åt andra hållet. Om människan vill att roboten ska utföra ett arbete så måste den få instruktioner som den kan förstå. Man skulle kunna säga att HMI-programmet fungerar som en tolk för att kunna möjliggöra kommunikation mellan människa och maskin. [1]

Typiska exempel på vad som kan finnas med i ett HMI är parametrar av olika slag. Maskinen visar de aktuella värdena och så kan man eventuellt ändra dessa med hjälp av gränssnittet så att man får de önskade värdena.

För att designa och programmera ett HMI-program kan man använda sig utav olika sorters mjukvaror. Ett exempel är Beijer Electronics mjukvara som heter iX Developer.

### 2.2 PLC

PLC står för Programmable Logic Controller och kan förklaras som ett programmerbart styrsystem. Det främsta användningsområdet för en PLC är inom industriell automation där den kan användas för att tala om för robotar eller annan automationsutrustning vad de ska göra. Det finns olika programmeringsspråk som används vid PLC-programmering. PLCn kopplas ofta ihop med ett HMI- eller SCADA-system för operatören enkelt ska kunna styra systemet. [2]

### 2.3 Mjukvaran

Mjukvaran som företaget tillhandahåller för att skapa HMI-programmet är iX Developer (version 2.40 SP5). Där används både C# scripting och .NET komponenter för att designa och specialisera sitt HMI till just det ändamål som önskas. Det finns många färdiga funktioner som kan vara bra att använda sig av när man ska skapa sitt HMI. Några exempel är möjligheten att skapa recept, ett alarmsystem och ett säkerhetssystem. [3]

För att programmera PLCn användes programmet GX Works3. Programmet stödjer alla huvudsakliga IEC språk. Det betyder att man kan välja mellan Ladder Diagram (LD), Function Block Diagram (FBD), Sequential Function Chart (SFC), Instruction List (IL) och Structured Text (ST). Det är även möjligt att använda olika språk inom samma projekt. [4]

Programmeringen av roboten skedde i Robot Studio. Med hjälp av programmet kan man skapa en 3D modell av sin robot och sedan programmera och simulera den offline. [5]

## 2.4 Hårdvaran

Panelen som ska visa HMI-programmet är ifrån Beijer Electronics och heter X2 base 10 v2 HP. Det är en TFT-LCD (Thin Film Transistor – Liquid Crystal Display) skärm med touch och den är 10'' stor vilket ger den 1024x600 pixlar. [6]

PLCn kommer ifrån Mitsubishi och tillhör MELSEC iQ-F serien. Det är en förbättrad variant av MELSEC-F serien med en snabbare bus, fler inbyggda funktioner och förbättrad utvecklingsmiljö i GX Works3. [7]

Roboten är ifrån ABB och är av modellen IRB 4600. Det är en robot som fungerar till flera olika ändamål. Det är en av de kompakteste robotarna av dess slag vilket leder till att den fungerar bra även om cellen den opererar i är relativt liten. [8]

## 3. Metod

I följande kapitel beskrivs arbetsprocessen under arbetet. Förberedelserna och informationshämtningen beskrivs här. Sen beskrivs hur HMI-programmet ska byggas upp samt vilka funktioner som finns färdiga i mjukvaran som skulle kunna användas. Det kommer även föras en diskussion kring varför källorna som använts i arbetet anses pålitliga.

### 3.1 Förberedelser

Allt började med ett uppstartsmöte om projektet via Microsoft teams eftersom flera på företaget var hemma i karantän. Då var det genomgång om vilka funktioner roboten skulle ha samt hur tanken var att de skulle uppnås.

Därefter fanns det tid att bekanta sig med den mjukvara som skulle användas under arbetet. Då arbetets fokus ligger på att skapa ett HMI-program som sedan går att återanvända, så lades mycket tid på att lära sig iX Developer. Först gick alla tutorials igenom på deras hemsida och sedan en snabb genomgång av handledaren på företaget där han visade saker som kan vara bra att känna till. Samtidigt gick det igenom vad HMI-programmet skulle innehålla i stora drag. Det fanns även möjlighet att se hur deras tidigare HMI-lösningar sett ut.

### 3.2 Programmering

Utifrån genomgången av iX Developer och inspiration från tidigare HMI-program så påbörjades HMI-programmeringen. Det var många funktioner som skulle implementeras men den mest centrala av dessa var möjligheten att bestämma hur många lådor det skulle vara på en pall och var de skulle placeras. Då man eventuellt vill stapla olika produkter på en pall men ändå använda samma robot så krävdes det att lådans dimensioner samt placering på pallen var ändringsbara.

#### 3.2.1 Taggar

I iX Developer så använder man sig utav taggar som kan kopplas till PLCn. Man kan välja vilken data typ taggen ska vara (INT16, BOOL, String, m.m.) och eventuellt vilken port på PLCn den ska kopplas till. Det kan vara uträkningar som sker i PLCn eller I/Os som den får in. Sen kopplas taggarna ihop med olika objekt eller specifika egenskaper hos objekten i HMI-programmet. Exempel på taggar syns i figur 4.1 och 4.2 under kapitel 4.1.

## 3.2.2 Lådans dimensioner

Eftersom man inte alltid har samma längd och bredd på lådorna när man kör roboten så krävs det att dessa går att ändra i HMI-programmet. Tanken är att man ska kunna ha olika mått på två olika lådor på samma lav men att deras höjd alltid är den samma. Det leder till att varje låda måste ha en egen tagg för längd och bredd men har en gemensam tagg för höjden.

Denna funktion är inget som behövs till det nuvarande projektet men kan behövas i framtida staplingslösningar.

## 3.2.3 Lådans position på pallen

En annan viktig funktion i HMI-programmet är när man ska placera lådorna på pallen. För att visualisera lådorna kommer knapparna att användas. Knapparnas längd och bredd kommer bero på de inmatade värdena och när man trycker på dessa så kan man ändra position i x- och y-led samt rotera dem, se figur 5.19 under kapitel 5.1.3. Rotationen används för att bestämma från vilket håll som roboten ska lämna lådan på pallen. Även här kräver det att varje låda har taggar som beskriver deras position i x- och y-led, rotation och deras position och storlek på panelen.

Funktionen är viktig för det skalbara programmet eftersom man förmodligen kommer använda ett annat staplingsmönster vid nästa projekt.

## 3.2.4 Recept

När alla lådornas dimensioner samt deras position på pallen är bestämda och inmatade i PLCn så ska dessa inställningar kunna sparas så att man i framtiden slipper ställa in roboten igen. Det går att göra via iX Developers receptfunktion. Där bestämmer man vilka taggar som ska ingå i receptet och de aktuella värdena på dessa sparas när man väljer att skapa ett nytt recept. När receptet sedan laddas så sätts taggarna till de sparade värdena. Funktionen tillåter maximalt 256 taggar. Se kapitel 5.1.3 för mer information om recept.

## 3.2.5 Skärmar

HMI-programmet består utav ett antal skärmar som visas beroende på vilken knapp man trycker på. Det finns en hemskärm där man kan skapa, ladda och radera recept. På hemskärmen kommer det även synas hur det nuvarande staplingsmönstret ser ut och några andra inställningar. Där finns även en översiktsskärm vilket visar en översikt på roboten när den är i drift.



### 3.3 Inhämtning av information

Mycket information är hämtad ifrån manualer som Beijer Elelectronics skapat. Där beskrivs alla tänkbara funktioner som kan användas i iX Developer samt några enkla kodexempel på lösningar. Annars har arbetet inte krävt så mycket informationshämtning.

### 3.4 Källkritik

Källorna som används under hela arbetet anses vara pålitliga. Antingen så är de ”peer reviewed” vilket betyder att de är granskade och accepterade av experter på ämnet. Annars så kommer källan ifrån företaget som tillhandahåller produkten i fråga. Det man får tänka på är att de inte lär ta upp alla negativa saker om produkten utan lägger bara fokus på de positiva.

## 4. Analys

I analyskapitlet kommer alla val som gjorts under arbetets gång att förklaras. Dessutom kommer en del problem som uppstått att tas upp samt hur lösningen på dessa blev.

### 4.1 HMI

Då HMI-programmet ska kunna återanvändas så är det viktigt att de olika parametrarna inte hårdprogrammeras. Till exempel så ska det vara möjligt att ändra lådornas dimensioner, positioner och rotation. Därför skapades det 9 taggar per låda som kopplades till PLCn. De organiserades enligt följande bild nedan.

D_Palleteringsdata	Word [Signed](0..4999)	D3000	Låda 1 Lav 1
D_P1_L1PosX	Word [Signed]	D3000	Låda 1 Lav 1
D_P1_L1PosY	Word [Signed]	D3001	Låda 1 Lav 1
D_P1_L1PosRZ	Word [Signed]	D3002	Låda 1 Lav 1
D_P1_L1Längd	Word [Signed]	D3003	Låda 1 Lav 1
D_P1_L1Bredd	Word [Signed]	D3004	Låda 1 Lav 1
D_P1_L1HMI_Left	Word [Signed]	D3005	Låda 1 Lav 1
D_P1_L1HMI_Top	Word [Signed]	D3006	Låda 1 Lav 1
D_P1_L1HMI_Width	Word [Signed]	D3007	Låda 1 Lav 1
D_P1_L1HMI_Height	Word [Signed]	D3008	Låda 1 Lav 1
D_P2_L1PosX	Word [Signed]	D3010	Låda 2 Lav 1
D_P2_L1PosY	Word [Signed]	D3011	Låda 2 Lav 1
D_P2_L1PosRZ	Word [Signed]	D3012	Låda 2 Lav 1
D_P2_L1Längd	Word [Signed]	D3013	Låda 2 Lav 1
D_P2_L1Bredd	Word [Signed]	D3014	Låda 2 Lav 1
D_P2_L1HMI_Left	Word [Signed]	D3015	Låda 2 Lav 1
D_P2_L1HMI_Top	Word [Signed]	D3016	Låda 2 Lav 1
D_P2_L1HMI_Width	Word [Signed]	D3017	Låda 2 Lav 1
D_P2_L1HMI_Height	Word [Signed]	D3018	Låda 2 Lav 1

*Figur 4.1. Variabler för låda 1 och 2 i lav 1.*

D_P1_L2PosX	Word [Signed]	D3000	Låda 1 Lav 2
D_P1_L2PosY	Word [Signed]	D4001	Låda 1 Lav 2
D_P1_L2PosRZ	Word [Signed]	D4002	Låda 1 Lav 2
D_P1_L2Längd	Word [Signed]	D4003	Låda 1 Lav 2
D_P1_L2Bredd	Word [Signed]	D4004	Låda 1 Lav 2
D_P1_L2HMI_Left	Word [Signed]	D4005	Låda 1 Lav 2
D_P1_L2HMI_Top	Word [Signed]	D4006	Låda 1 Lav 2
D_P1_L2HMI_Width	Word [Signed]	D4007	Låda 1 Lav 2
D_P1_L2HMI_Height	Word [Signed]	D4008	Låda 1 Lav 2
D_P2_L2PosX	Word [Signed]	D4010	Låda 2 Lav 2
D_P2_L2PosY	Word [Signed]	D4011	Låda 2 Lav 2
D_P2_L2PosRZ	Word [Signed]	D4012	Låda 2 Lav 2
D_P2_L2Längd	Word [Signed]	D4013	Låda 2 Lav 2
D_P2_L2Bredd	Word [Signed]	D4014	Låda 2 Lav 2
D_P2_L2HMI_Left	Word [Signed]	D4015	Låda 2 Lav 2
D_P2_L2HMI_Top	Word [Signed]	D4016	Låda 2 Lav 2
D_P2_L2HMI_Width	Word [Signed]	D4017	Låda 2 Lav 2
D_P2_L2HMI_Height	Word [Signed]	D4018	Låda 2 Lav 2

*Figur 4.2. Variabler för låda 1 och 2 i lav 2.*

Det öppnar upp möjligheten att stapla 100 lådor per lav vilket förmodligen är mycket mer än vad man någonsin kommer behöva.

För att välja önskad position på lådorna samt visualisera den på touchpanelen så valdes följande metod. Man börjar med att mata in antalet önskade lådor samt deras längd och bredd. Därefter kommer nästa skärm att visa numrerade knappar vilket då representerar de lådorna som matades in på föregående skärm. När knapparna (lådorna) trycks ned så kan önskade positioner och rotation ställas in.

Från början så var tanken att man skapade två olika staplingsmönster och att de kördes vart annat lav. Då detta kanske inte alltid är önskat så skapades skärmen nedan. Genom att trycka på ett lav så ändras det vilket staplingsmönster som ska köras där.



Figur 4.3. Val av lavmönster.

## 4.2 Problem

Vid skapandet av HMI-programmet har det uppstått en del problem på vägen. Det första och kanske största problemet var att man inte kan rotera objekt i iX Developer. Då det bara skulle vara möjligt att rotera lådorna med en vinkel på 0, 90, 180 och 270 grader så löstes det genom att ändra knapparna längd och bredd på skärmen beroende på vilken vinkel som matats in.

Visualiseringen av rotationsvinkeln skulle ske genom en pil som roterade beroende på taggens värde eftersom det annars inte varit möjligt att se om en låda var till exempel vinklad 0 eller 180 grader. Men då knappen som representerar lådan ändrar storlek från knappt synlig till de värden som matats in så kommer bilden inte förstöras. Däremot så gick det att skala text på knappen beroende på en taggs värde. Lösningen blev då att representera de fyra vinklarna med de här symbolerna:

$> = 0$  grader

$V = 90$  grader

$< = 180$  grader

$\wedge = 270$  grader

Ett annat problem som uppstod var att synligheten av objekt, till exempel knappar, gick bara att ställa in till att vara synlig för endast ett värde på en tagg, annars så var den osynlig. Då en rotation under 0 grader och över 270 grader inte ska vara möjlig så vill man att operatören inte ska kunna gå utanför det intervallet. Tanken var att operatören inte ska kunna mata in ett eget värde utan bara ändra vinkeln med hjälp av två knappar som antingen ökade eller minskade

rotationsvinkeln med 90 grader när någon av dem klickades. Skulle man redan ha en rotationsvinkel på 270 grader så skulle ökningsknappen försvinna eller inte gå att trycka på. Eftersom synlighetsinställningarna såg ut som de gjorde blev lösningen att tre knappar sattes på varandra för både ökning och minskning. De minskande knapparna är synliga vid 90, 180 och 270 grader och de ökande syns vid 0, 90 och 180 grader. Då ett tryck på en knapp motsvarar en ökning eller minskning med 90 grader så kommer operatören aldrig lyckas mata in något annat värde än det som är tänkt.

## 5. Resultat

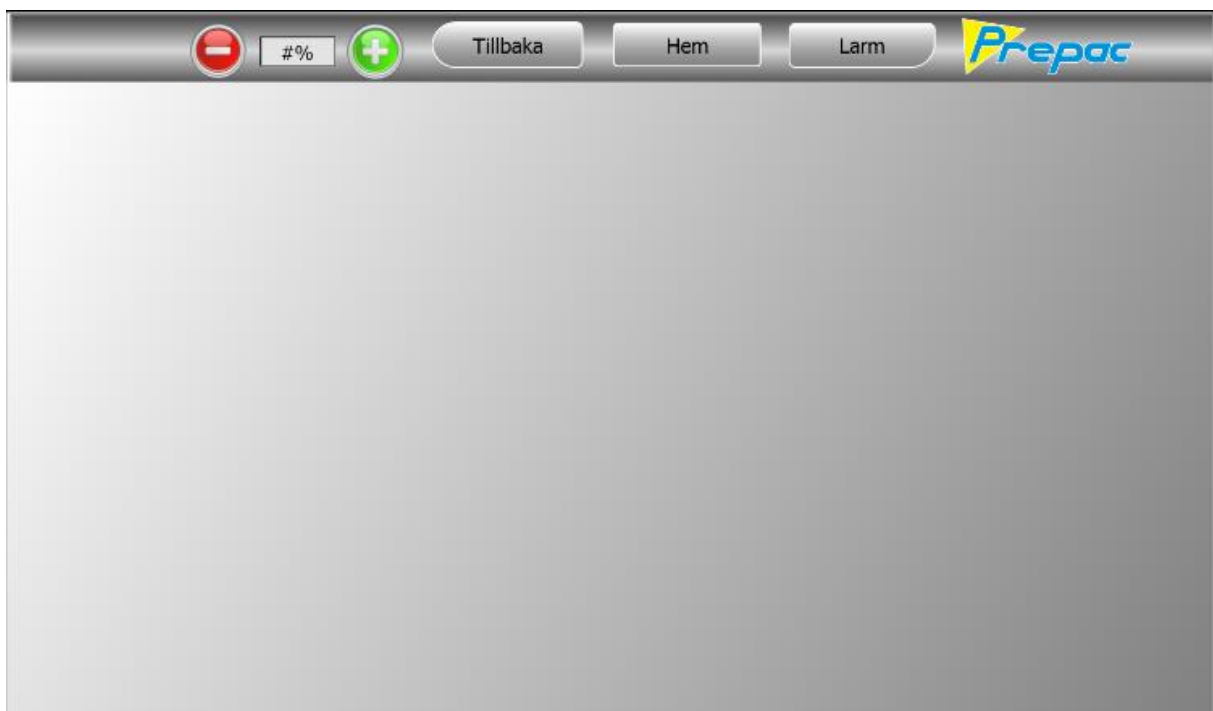
Resultatet går igenom hur hela det slutgiltiga HMI-programmet ser ut och alla dess funktioner. Det förklaras även hur lösningarna bakom alla funktioner ser ut.

### 5.1 HMI

HMI-programmets uppgifter är att hantera recept (dvs skapa, ändra, ladda och radera dessa) och visualisera körningen i realtid. Det kommer även larma om det är något som går fel. För att göra programmet så användarvänligt som möjligt så har antalet skärmar minimerats för att enkelt navigera sig. Dessutom har alla tänkbara fel, som kan uppstå på grund av felaktig inmatning eller liknande av operatören, eliminerats.

#### 5.1.1 Bakgrund

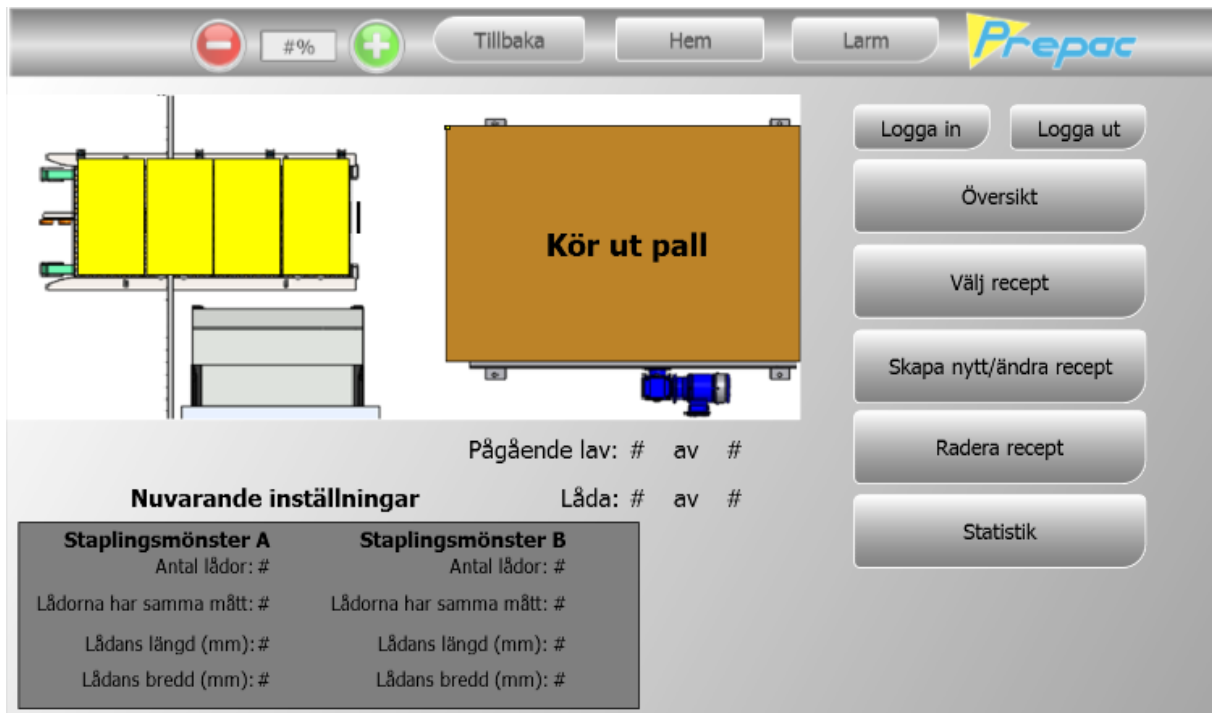
Bakgrunden är en skärm som ligger bakom alla de andra skärmarna. Den består utav 3 knappar, hem, tillbaka och larm. ”Hem”-knappen visar den skärm som är satt som hemskärm, ”Tillbaka”-knappen visar föregående skärm och ”Larm”-knappen visar alarmlistan. Här finns även möjlighet att se och ställa in med vilken hastighet man ska köra roboten. Till sist så kan man se företagets logga uppe i det ena hörnet.



*Figur 5.1. "Bakgrunds"-skärm*

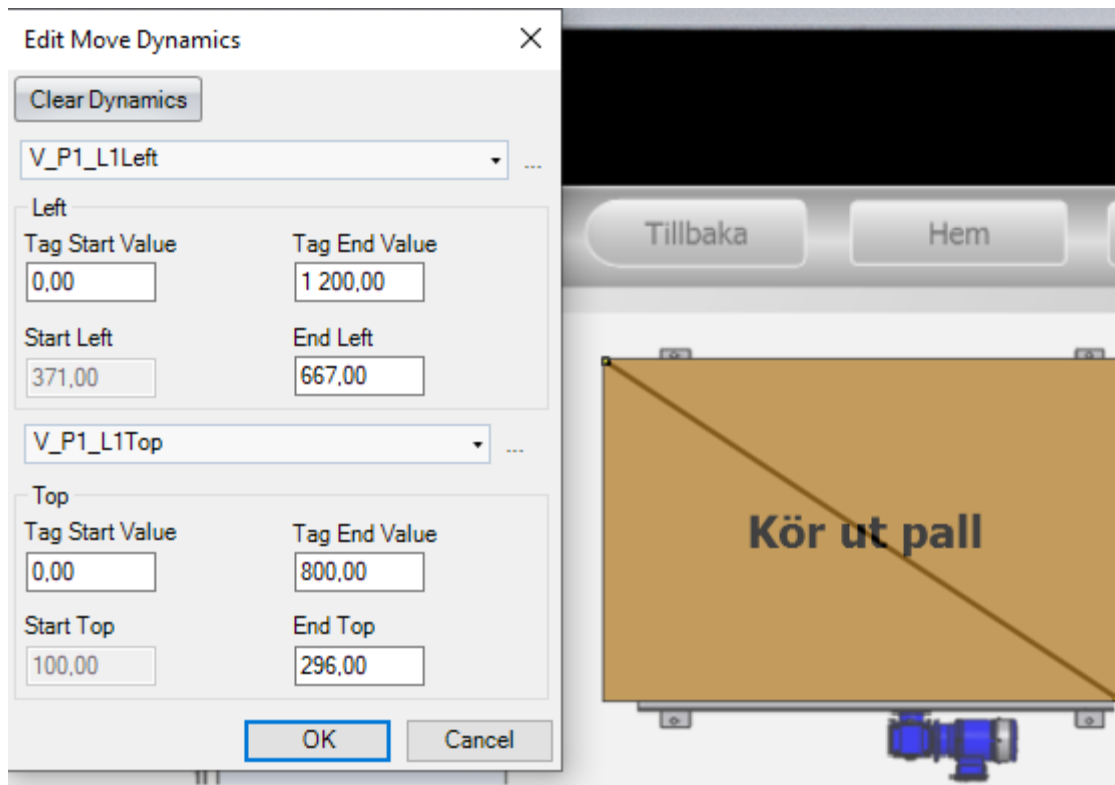
## 5.1.2 Visualisering

Visualiseringsskärmen är även programmets hemskärm och den visar hur många lådor som är staplade samt hur de är staplade på det aktuella laget. De nuvarande inställningarna visas i en ruta nere till höger. På skärmen finns även knappar för att ta sig igenom alla de andra skärmarna i programmet.

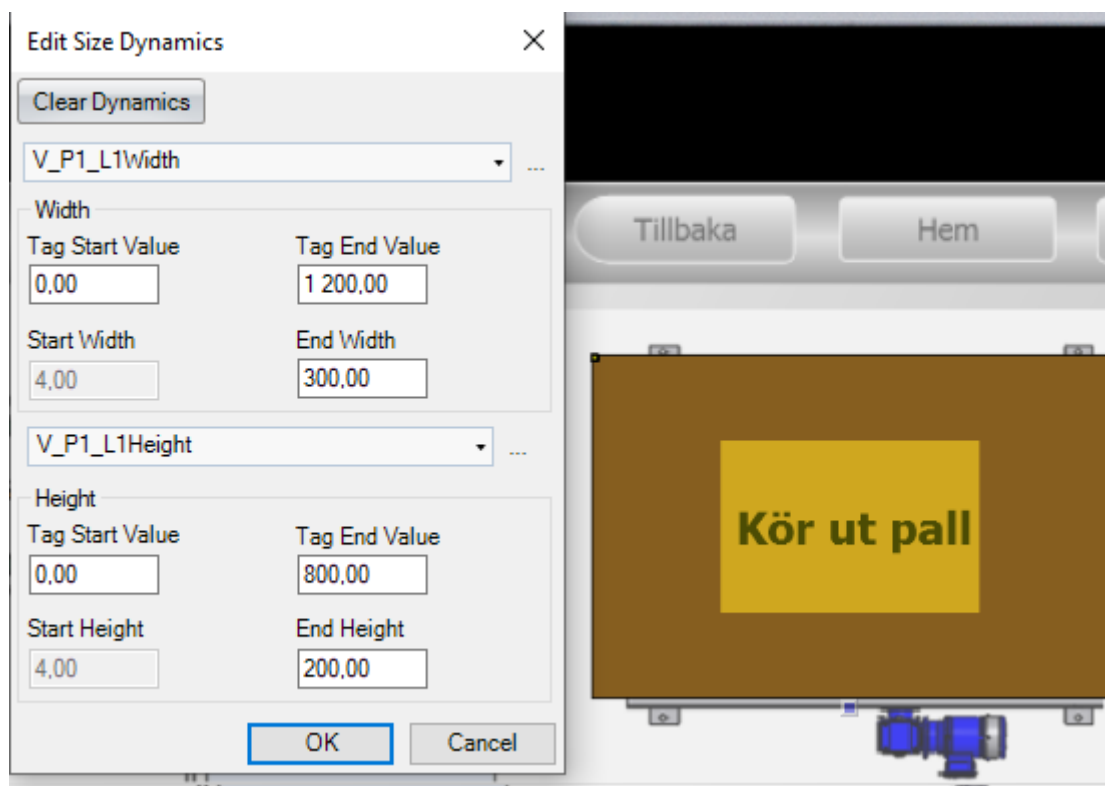


Figur 5.2. "Visualiserings"- och "Hem"-skärmen

När maskinen är i drift ska lådorna placeras på skärmens pall enligt det staplingsmönster som roboten placerar de verkliga lådorna. För att lyckas med det så används "move" och "size" egenskaperna hos rektangel objekten i iX Developer. Rektanglarnas utgångsposition är uppe i pallens vänstra hörn och beroende på den verkliga lådans dimensioner och position så kommer dessa flyttas och ändra storlek.



Figur5.3. "Move" funktionen på ett objekt



Figur 5.4. "Size" funktionen på ett objekt.

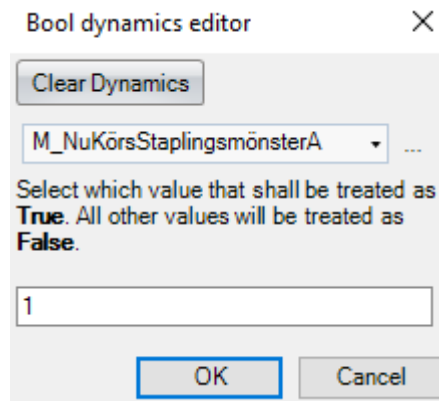


Det krävs en skalning av de värden som matats in för att få en realistisk bild på panelen. För att åstadkomma denna användes script i iX Developer som uppdaterar lådornas storlek och position varje gång som visualiseringsskärmen öppnas. Koden upprepades för varje låda i alla staplingsmönster.

```
if(Globals.Tags.D_AntallLådor_StaplingsmönsterA.Value > 0) //Kollar hur många lådor som ska visas (SMA).
{
    if(Globals.Tags.D_P1_LlPosRZ.Value == 0) //Kollar hur lådan är vinklad för att korrekt kunna justera längd och bredd.
    {
        Globals.Tags.V_P1_LlHeight.Value = Globals.Tags.D_P1_LlHMI_Height.Value;
        Globals.Tags.V_P1_LlWidth.Value = Globals.Tags.D_P1_LlHMI_Width.Value;
        Globals.Tags.V_P1_LlLeft.Value = Globals.Tags.D_P1_LlPosY.Value;
        Globals.Tags.V_P1_LlTop.Value = Globals.Tags.D_P1_LlPosX.Value - (Globals.Tags.D_P1_LlLängd.Value/2);
    }
    else if(Globals.Tags.D_P1_LlPosRZ.Value == 90)
    {
        Globals.Tags.V_P1_LlHeight.Value = Globals.Tags.D_P1_LlHMI_Height.Value;
        Globals.Tags.V_P1_LlWidth.Value = Globals.Tags.D_P1_LlHMI_Width.Value ;
        Globals.Tags.V_P1_LlLeft.Value = Globals.Tags.D_P1_LlPosX.Value - (Globals.Tags.D_P1_LlLängd.Value/2);
        Globals.Tags.V_P1_LlTop.Value = Globals.Tags.D_P1_LlPosY.Value;
    }
    else if(Globals.Tags.D_P1_LlPosRZ.Value == 180)
    {
        Globals.Tags.V_P1_LlHeight.Value = Globals.Tags.D_P1_LlHMI_Height.Value;
        Globals.Tags.V_P1_LlWidth.Value = Globals.Tags.D_P1_LlHMI_Width.Value;
        Globals.Tags.V_P1_LlLeft.Value = Globals.Tags.D_P1_LlPosY.Value - (Globals.Tags.D_P1_LlBredd.Value);
        Globals.Tags.V_P1_LlTop.Value = Globals.Tags.D_P1_LlPosX.Value - (Globals.Tags.D_P1_LlLängd.Value/2);
    }
    else if(Globals.Tags.D_P1_LlPosRZ.Value == 270)
    {
        Globals.Tags.V_P1_LlHeight.Value = Globals.Tags.D_P1_LlHMI_Height.Value;
        Globals.Tags.V_P1_LlWidth.Value = Globals.Tags.D_P1_LlHMI_Width.Value;
        Globals.Tags.V_P1_LlLeft.Value = Globals.Tags.D_P1_LlPosY.Value - (Globals.Tags.D_P1_LlLängd.Value/2);
        Globals.Tags.V_P1_LlTop.Value = Globals.Tags.D_P1_LlPosX.Value - (Globals.Tags.D_P1_LlBredd.Value);
    }
}
else //Annars ska lådan inte visas.
{
    Globals.Tags.V_P1_LlHeight.Value = 0;
    Globals.Tags.V_P1_LlWidth.Value = 0;
    Globals.Tags.V_P1_LlLeft.Value = 0;
    Globals.Tags.V_P1_LlTop.Value = 0;
}
```

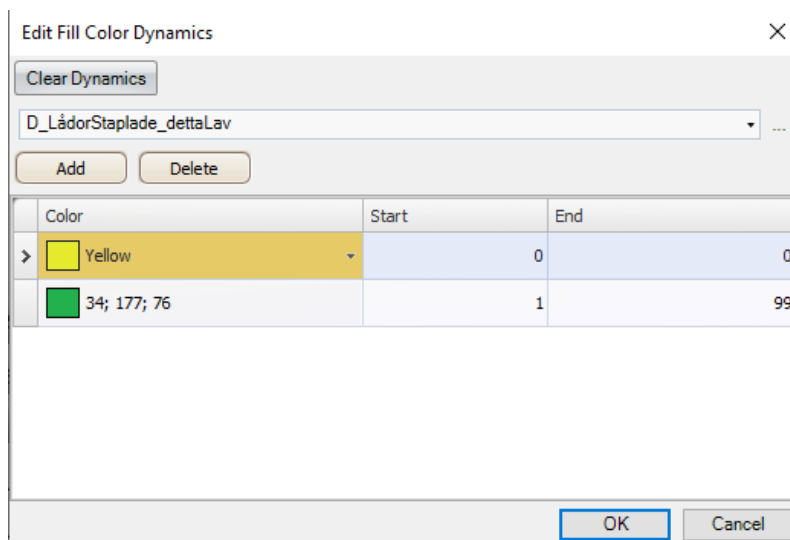
##### *5.5. Kod som bestämmer hur stor låda 1 i lav 1 ska vara samt var den ska befinna sig på visualiseringsskärmen.*

Då all scripting i iX Developer är eventbaserad så måste alla lådor i alla lav placeras på skärmen när den öppnas. Alternativet hade varit att skapa en knapp som man manuellt måste trycka för att uppdatera skärmen vilket inte skulle vara speciellt praktiskt. Lösningen blev i stället att alla lådor läggs ut när skärmen startar men beroende på vilket lav man befinner sig på så visas bara de aktuella lådorna. Så när ett lav som ska köra staplingsmönster A körs så visas lådorna för de staplingsmönstret, annars är de osynliga.



Figur 5.6. "Visibility" funktionen för ett objekt.

För att visualisera att lådan lagts på pallen så ändras färgen från gul till grön. Taggen "D\_LådorStaplade\_dettaLav" håller koll på hur många lådor som staplats på det aktuella lavet. Så om den är större än 0 så ska första lådan bli grön, om den är större än 1 ska den andra lådan bli grön osv.



Figur 5.7. "Fill color" funktionen för ett objekt.

Samma princip gäller för de fyra köplatserna för lådorna som syns längst till vänster på skärmen. Rektanglarna byter färg från gult till grönt när givaren känner av att där befinner sig en låda på någon av platserna.

Skärmen visar även hur många lav och totalt antal lådor som ska staplas per pall samt vilken som är den aktuella lådan och lavet. Antalet lav per pall har matats in av operatören men totala antalet lådor kan variera beroende på hur många gånger man väljer att köra de olika staplingsmönsterna. Därför sker det en beräkning när skärmen öppnas som ser ut enligt följande.

```

void Visualisering_Opened(System.Object sender, System.EventArgs e)
{
    int AntalLav = Globals.Tags.D_AntalLavPerPall.Value;
    int LådorSMA = Globals.Tags.D_AntalLådor_StaplingsmönsterA.Value;
    int LådorSMB = Globals.Tags.D_AntalLådor_StaplingsmönsterB.Value;
    int AntalLavSMA = 0;
    int AntalLavSMB = 0;

    if(AntalLav > 0) //Räknar hur många lav SMA respektive SMB.
    {
        if(Globals.Tags.B_StaplingsmönsterLav1.Value == 1)
        {
            AntalLavSMB = AntalLavSMB + 1;
        }
        else
        {
            AntalLavSMA = AntalLavSMA + 1;
        }
    }
    if(AntalLav > 1)
    {
        if(Globals.Tags.B_StaplingsmönsterLav2.Value == 1)
        {
            AntalLavSMB = AntalLavSMB + 1;
        }
        else
        {
            AntalLavSMA = AntalLavSMA + 1;
        }
    }
}

```

Figur 5.8. Räknar hur många lav som kör staplingsmönster A respektive staplingsmönster B.

```

if(AntalLav > 15)
{
    if(Globals.Tags.B_StaplingsmönsterLav16.Value == 1)
    {
        AntalLavSMB = AntalLavSMB + 1;
    }
    else
    {
        AntalLavSMA = AntalLavSMA + 1;
    }
}

Globals.Tags.D_AntalLådorPerPall.Value = AntalLavSMA * LådorSMA + AntalLavSMB * LådorSMB; //Räknar antal låder per pall.

```

Figur 5.9. Uträkning av antalet lådor per pall.

I inställningsrutan så syns de olika staplingsmönsterna med hur många lådor som befinner sig i varje mönster. Det visas även om alla lådorna i ett staplingsmönster har samma längd och bredd och skulle så vara fallet så skrivs även dessa ut, annars så syns de inte. För att bestämma om alla lådor i ett staplingsmönster har samma dimensioner så kollas det först hur många lådor det är i lavet. Efter det jämförs längden och bredden på låda 1 med längden och bredden på låda 2. Är de samma så jämförs det med låda 3 osv. Skulle alla jämförelser vara ok så har alla lådor samma mått.

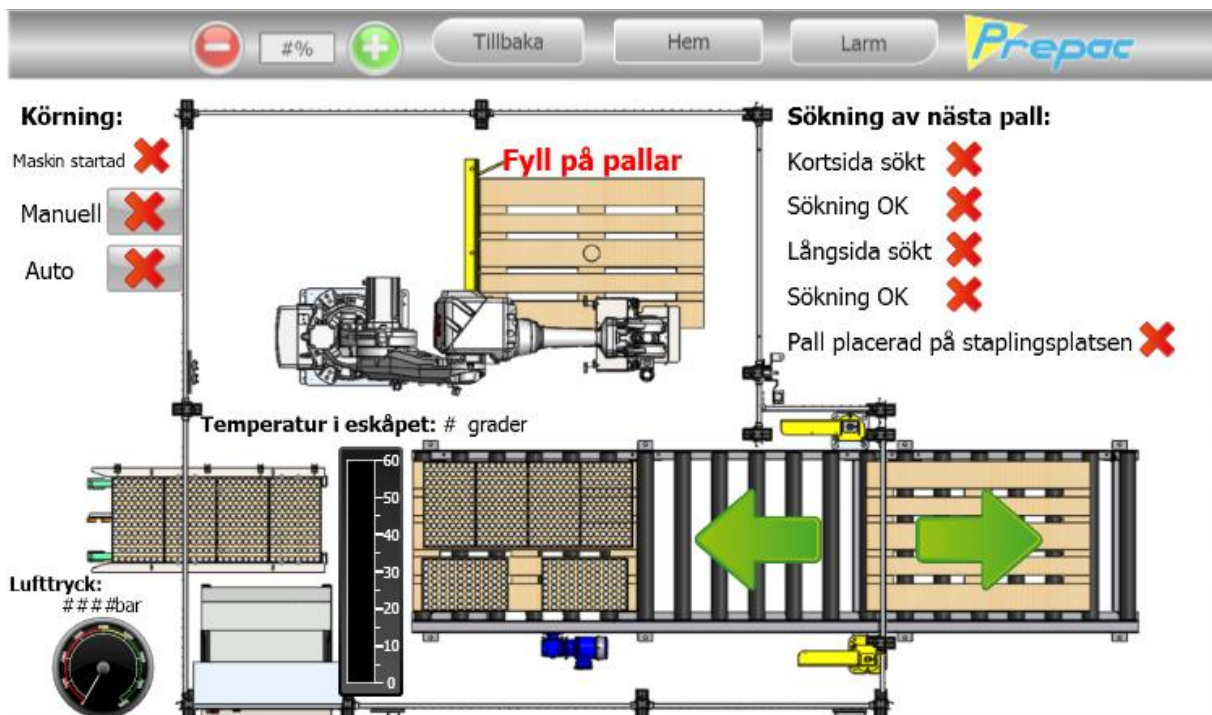
## 5.1.2 Översikt

Översiktsskärmen visar en översikt över hela cellen som roboten befinner sig i. Där går det att byta mellan manuell och automatisk körning. När den befinner sig i manuell körning så går det att köra pallen fram och tillbaka på rullbandet med hjälp av de gröna pilarna som dyker upp, annars så kör den automatiskt ut pallen när den är full. Knapparna som pilarna sitter på fungerar så att medan man håller inne knappen så sätter man taggen som styr bandet till 1 och när man släpper knappen så går den tillbaka till 0.



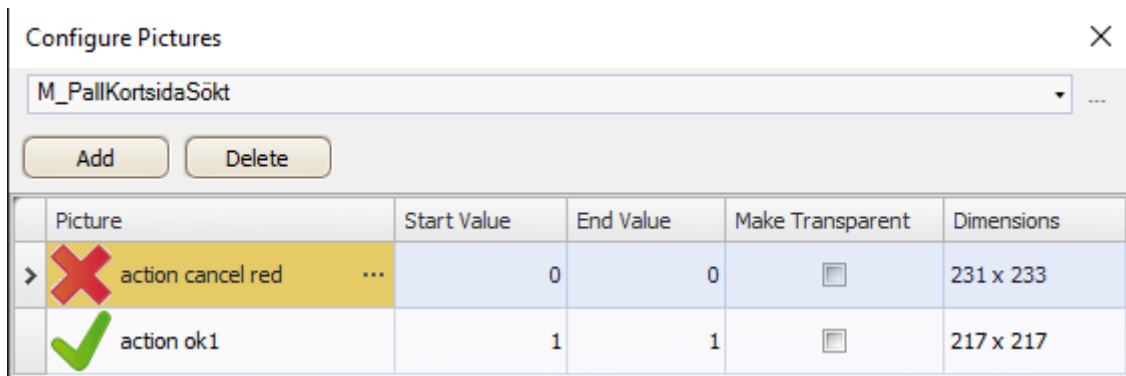
Figur 5.10. Set / Reset tag för transportörbandet

På skärmen går det även att se hur pallstatusen ser ut. För att roboten ska kunna hitta och hämta pallen från pallhögen och ställa på rullbandet så krävs det 2 sökningar. Det går att se om sökningarna är gjorda och om de blev godkända. Roboten kan även känna av om pallhögen är tom vilket då startar en blinkande text att det är dags för påfyllning av paller. Det sker även en sökning när pallen är lämnad på rullbandet för att se om den är rätt placerad.

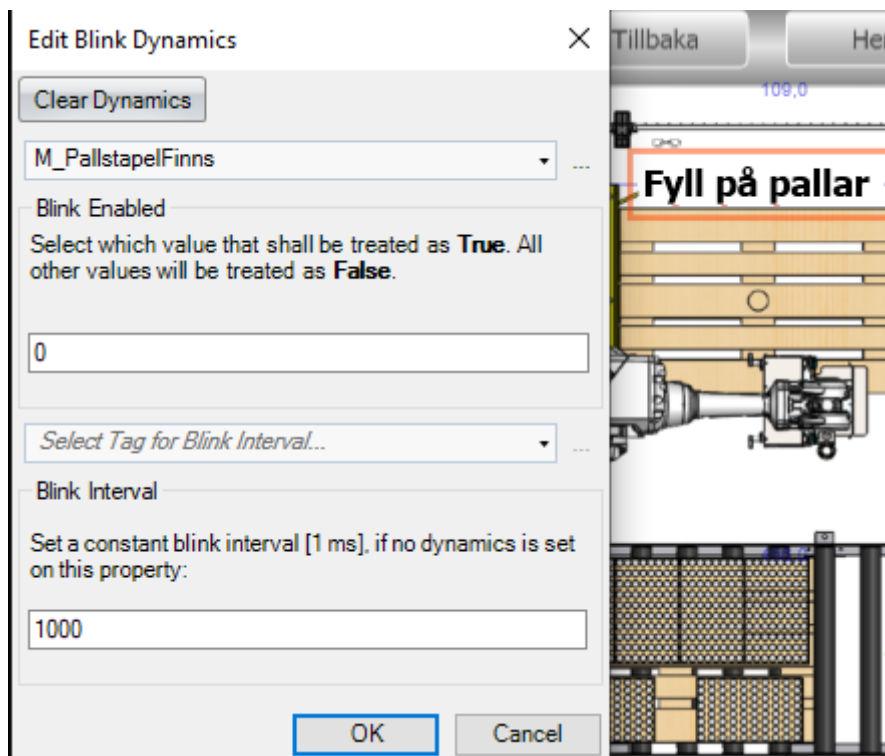


Figur 5.11. Översiktsskärm

För att visa att en sökning är genomförd eller att pallen är på plats kommer de röda kryssen ändras till en grön bock. Bilden kommer att ändras beroende på taggens värde.



Figur 5.12. "Multi picture" objekt i iX Developer



Figur 5.13. Blinkfunktionen för en text

### 5.1.3 Recept

I iX Developers receptfunktion kan man skapa ett recept med 256 taggar. När man väljer att spara recept kommer taggarnas nuvarande värde sparas i receptet. Det är sedan möjligt att ladda ett recept vilket då gör att taggarna får värdena som sparats i receptet. Alla lådors taggar med deras position och dimension samt hur lavmönstret ska se ut kommer sparas undan i recept.

"Välj recept"-knappen laddar ett sparat recept och "Radera recept"-knappen raderar ett sparat recept. "Skapa nytt/ändra recept"-knappen tar en igenom ett antal skärmar som låter användaren ändra värdena på de taggar som ingår i receptet innan man till sist sparar det.

### 5.1.3.1 Skapa recept

När man vill skapa ett nytt eller ändra på ett befintligt recept så tas man först till skärmen nedan. Där bestäms det hur många lav som ska finnas på pallan, hur många olika staplingsmönster som ska användas samt lådornas och pallens höjd.



Figur 5.14. "Skapa pall"-skärmen

Efter "Fortsätt"-knappen klickats öppnas nästa skärm där antalet lådor, och deras dimensioner, i staplingsmönster A ska bestämmas.



Figur 5.15. Skapa staplingsmönster A

När "OK"-knappen, bredvid antal lådor fältet, klickats så möjliggörs det att bestämma längd och bredd för så många lådor som matats in i fältet. Ett utdrag av koden som körs när ok knappen klickats syns nedan, figur 5.16.

```
public partial class rSkapaSMA
{
    void Button_Click(System.Object sender, System.EventArgs e) //När knappen "Ok" trycks så körs följande kod.
    {
        int nbr = Globals.Tags.D_AntalLådor_StaplingsmönsterA.Value; //Värdet av taggen "D_AntalLådor_udda" läggs i variabeln "nbr".

        if(nbr > 12)
        {
            MessageBox.Show("Max 12 lådor/lav", "Info"); //Om man försöker skriva in fler än 12 lådor så säger den ifrån.
        }
        else //Annars om nbr är mellan 0 och 12 så händer följande.
        {
            if(nbr > 0)
            {
                this.txtl0.Visible = true; //Om nbr är större än 0 så visas text och textfälten för låda 1.
                this.txtLängd10.Visible = true;
                this.txtBredd10.Visible = true;
                this.txtfLängd10.Visible = true;
                this.txtfBredd10.Visible = true;
            }
            else
            {
                this.txtl0.Visible = false; //Annars om nbr är mindre än eller lika med 0 så visas de inte.
                this.txtLängd10.Visible = false;
                this.txtBredd10.Visible = false;
                this.txtfLängd10.Visible = false;
                this.txtfBredd10.Visible = false;

                Globals.Tags.D_P1_L1Längd.Value = 0; //Värdena på längd och bredd sätts även till 0.
                Globals.Tags.D_P1_L1Bredd.Value = 0;
                //Detta görs för att rensa taggarna på gamla värden om man går från ett större antal lådor till ett lägre antal.
            }
        }
    }
}
```

Figur 5.16. Kod för att visa rätt antal lådor som ska få längd och bredd. Bilden visar endast kod för låda 1 men när "nbr > 1" så ändras textfälten för låda 2 från osynliga till synliga osv.

Om man önskar att alla lådor ska ha samma längd och bredd så kan man trycka på knappen alla lådor har samma mått, i stället för att mata in samma längd och bredd för varje låda. Det som sker då är att beroende på hur många lådor som ska finnas med i staplingsmönstret så sätts deras längd och bredd till samma värde som låda 1 har.

```
void btnALHSM_Click(System.Object sender, System.EventArgs e) //När knappen "Alla lådor har samma värde" trycks så körs följande kod.
{
    int nbr = Globals.Tags.D_AntalLådor_StaplingsmönsterA.Value; //"D_AntalLådor_udda"s värde läggs i variabeln "nbr".

    if(nbr > 1)
    {
        Globals.Tags.D_P2_L1Bredd.Value = Globals.Tags.D_P1_L1Bredd.Value; //Om nbr är större än 1 så kommer taggarna för låda2s längd och bredd
        Globals.Tags.D_P2_L1Längd.Value = Globals.Tags.D_P1_L1Längd.Value;
    }

    if(nbr > 2) //Samma kod upprepas för varje låda.
    {
        Globals.Tags.D_P3_L1Bredd.Value = Globals.Tags.D_P1_L1Bredd.Value;
        Globals.Tags.D_P3_L1Längd.Value = Globals.Tags.D_P1_L1Längd.Value;
    }
}
```

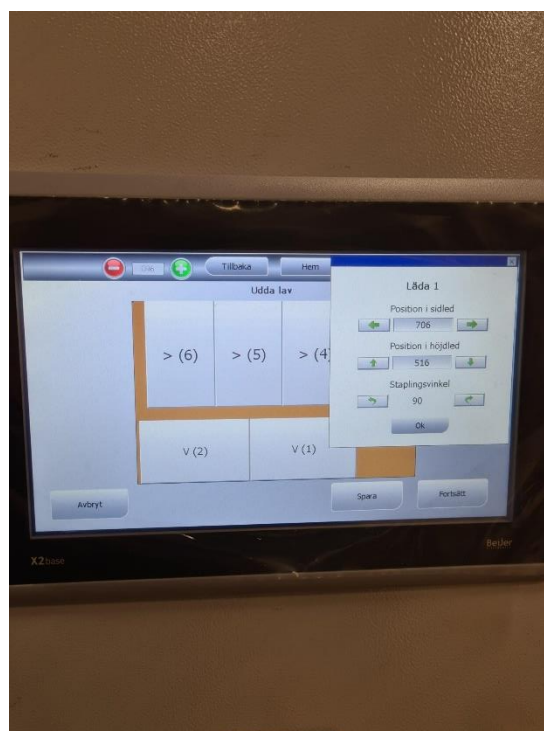
Figur 5.17. Kod som gör att alla lådor får samma mått

När dimensioneringen av alla lådor är klara så ska deras position i staplingsmönstret väljas. På nästa skärm syns en stor rektangel som ska föreställa pallen som lådorna ska staplas på.





Figur 5.18. Skärm för att skapa staplingsmönster A



Figur 5.19. Val av staplingsmönster A när roboten är igång.

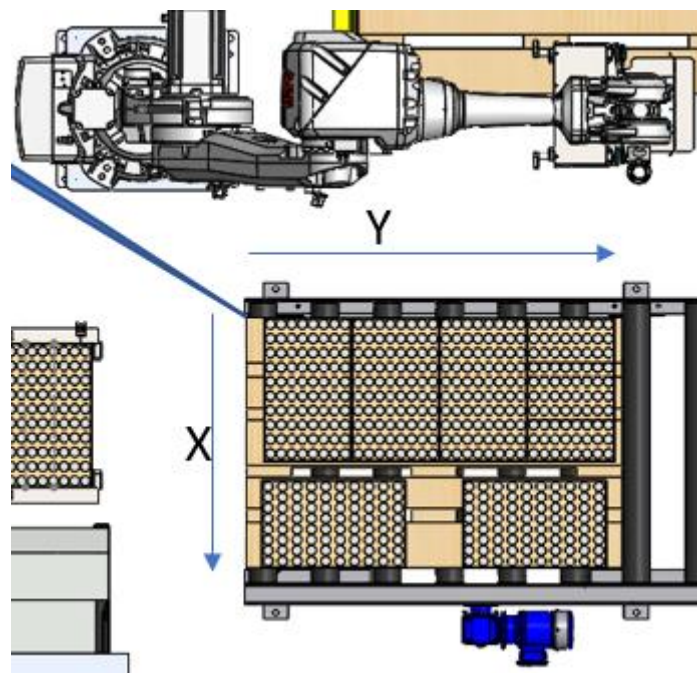
Lådorna representeras av knappar som ändrar storlek beroende på vilken längd och bredd som matats in på föregående skärm. Även antalet lådor som visas beror på vad som valdes tidigare. Knapparna som representerar lådorna är numrerade så att man ska kunna bestämma i vilken ordning de ska staplas på pallen. De visar också med vilken ingångsvinkel mot pallen som är inställd. Det görs eftersom det annars inte varit möjligt att se skillnad på 0 och 90 grader samt 180 och 270 grader, om man inte tryckt på knappen för att kolla värdet. När en knapp klickats visas följande pop-up fönster som möjliggör ändring av lådans position och vinkling.



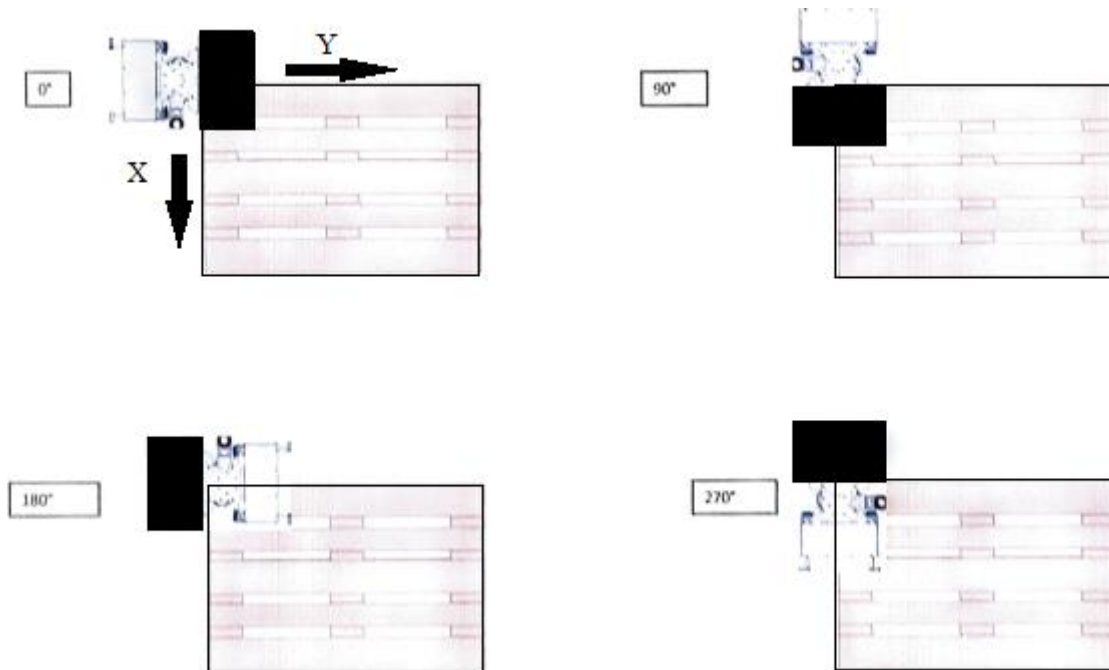


Figur 5.20. Pop-up fönster för att bestämma en lådas position.

Koordinaterna på pallen går ifrån (0, 0) uppe i det vänstra hörnet till (1200, 800) nere i det högra. Lådans koordinater utgår ifrån mitten av verktyget som hämtar lådan. Alltså hamnar den mitt på en av lådans sidor.



Figur5.21. X- och Y-koordinater för pallen



Figur 5.22. X- och Y-koordinater (0, 0) för lådan med alla möjliga vinklar

För att undvika att man sätter 2 lådor så att de överlappar så körs en script, som kollar hur alla lådor befinner sig, när man trycker på "Ok". Skulle den då överlappa en annan låda så dyker en varningstext upp. Figuren nedan visar kod när positionen för låda 1 jämförs med positionen för låda 2 för vinklarna 0 och 90 grader på låda.

```
//Kollar så att låda 1 inte ligger på Låda 2.
if(Globals.Tags.D_P2_LlPosRZ.Value == 0 && Globals.Tags.D_P1_LlPosRZ.Value == 0) //Låda 1 0 grader
{
    if(Globals.Tags.D_P1_LlPosY.Value < (Globals.Tags.D_P2_LlPosY.Value + Globals.Tags.D_P2_LlBredd.Value)
    && Globals.Tags.D_P1_LlPosY.Value > (Globals.Tags.D_P2_LlPosY.Value - Globals.Tags.D_P1_LlBredd.Value)
    && Globals.Tags.D_P1_LlPosX.Value < (Globals.Tags.D_P2_LlPosX.Value + Globals.Tags.D_P1_LlLängd.Value/2)
    && Globals.Tags.D_P1_LlPosX.Value > (Globals.Tags.D_P2_LlPosX.Value - Globals.Tags.D_P1_LlLängd.Value/2))
    {
        MessageBox.Show("Låda 1 befiiner sig på en annan låda", "Varning");
    }
    else
    {
        Globals.rSkapaSMapos.Show();
    }
}
else if(Globals.Tags.D_P2_LlPosRZ.Value == 90 && Globals.Tags.D_P1_LlPosRZ.Value == 0)
{
    if(Globals.Tags.D_P1_LlPosY.Value < (Globals.Tags.D_P2_LlPosY.Value + Globals.Tags.D_P2_LlLängd.Value/2)
    && Globals.Tags.D_P1_LlPosY.Value > (Globals.Tags.D_P2_LlPosY.Value - Globals.Tags.D_P1_LlBredd.Value)
    && Globals.Tags.D_P1_LlPosX.Value < (Globals.Tags.D_P2_LlPosX.Value + Globals.Tags.D_P1_LlLängd.Value/2)
    && Globals.Tags.D_P1_LlPosX.Value > (Globals.Tags.D_P2_LlPosX.Value - Globals.Tags.D_P1_LlLängd.Value/2))
    {
        MessageBox.Show("Låda 1 befiiner sig på en annan låda", "Varning");
    }
    else
    {
        Globals.rSkapaSMapos.Show();
    }
}
}
```

Figur 5.23. Utdrag från kod som kollar så att låda 1 inte ligger ovanpå låda 2 i lav 1.

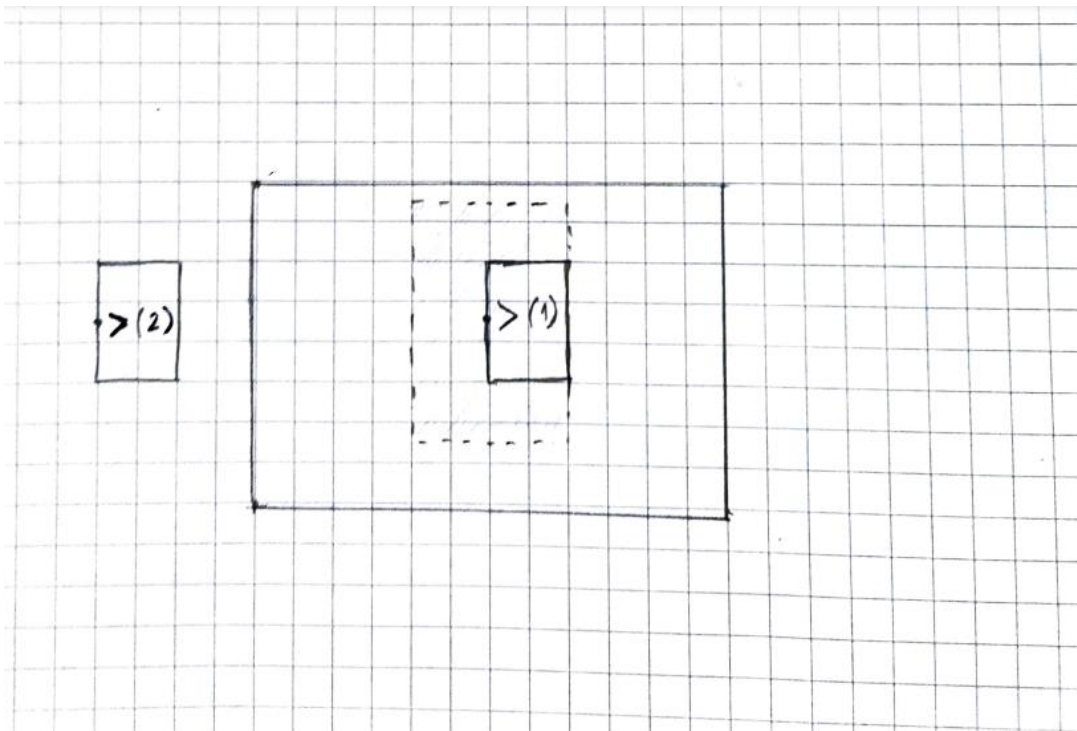
För att säkerhetsställa att lådan som ska placeras inte befinner sig på en annan låda så kollar man först vilken vinkel lådan som ska placeras har och vilken vinkel alla andra lådor har. Utifrån vinklarna går det att bestämma vilka positioner i koordinatsystemet som lådan inte får placeras på.

Låt säga att lådan som redan befinner sig på pallen är placerad med vinkeln 0 grader. Sen ska ytterligare en låda placeras, denna också med vinkeln 0 grader. Då gäller det att den andra lådans koordinater inte befinner sig innanför det streckade området för att undvika överlappning. Om utgångspunkten är den första lådans koordinater så kan man säga att ingen låda med 0 graders vinkel får ha koordinater som ligger i området:

$$L1.posY - L2.bredd > L2.posY > L1.posY + L1.bredd$$

$$L1.posX - L1.längd/2 - L2.längd/2 > L2.posX > L1.posX + L1.längd/2 + L2.längd/2$$

Lägg märke till att x-led är lodrät och y-led är vågrät i bilden nedan samt att punkten (0,0) ligger uppe i vänstra hörnet av den stora rektangeln.



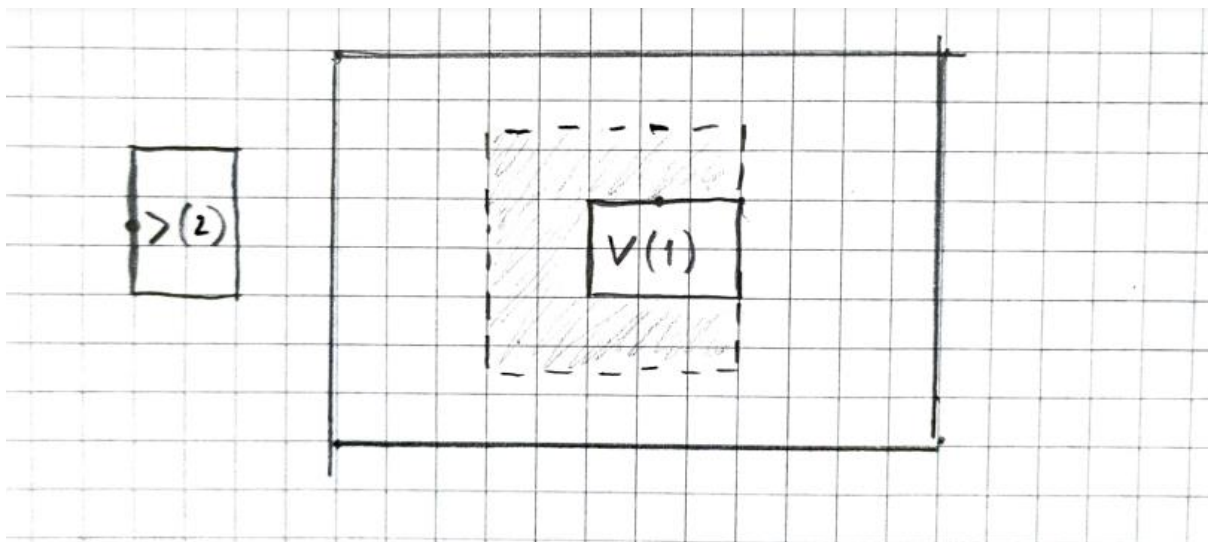
Figur 5.24. Exempel 1 på hur lådor inte ska överlappas.

Om den placerade lådan i stället har en vinkel på 90 grader och lådan som ska placeras fortfarande är med vinkeln 0 grader så ändras området där den inte får placeras. Den här gången gäller följande:

$$L1.posY - L1.längd/2 - L2.bredd < L2.posY < L1.posY + L1.längd/2$$

$$L1.posX - L2.längd/2 < L2.posX < L1.posX + L1.bredd + L2.längd/2$$

Notera att längden på låda 1 är vågrät nu eftersom lådan roterats.

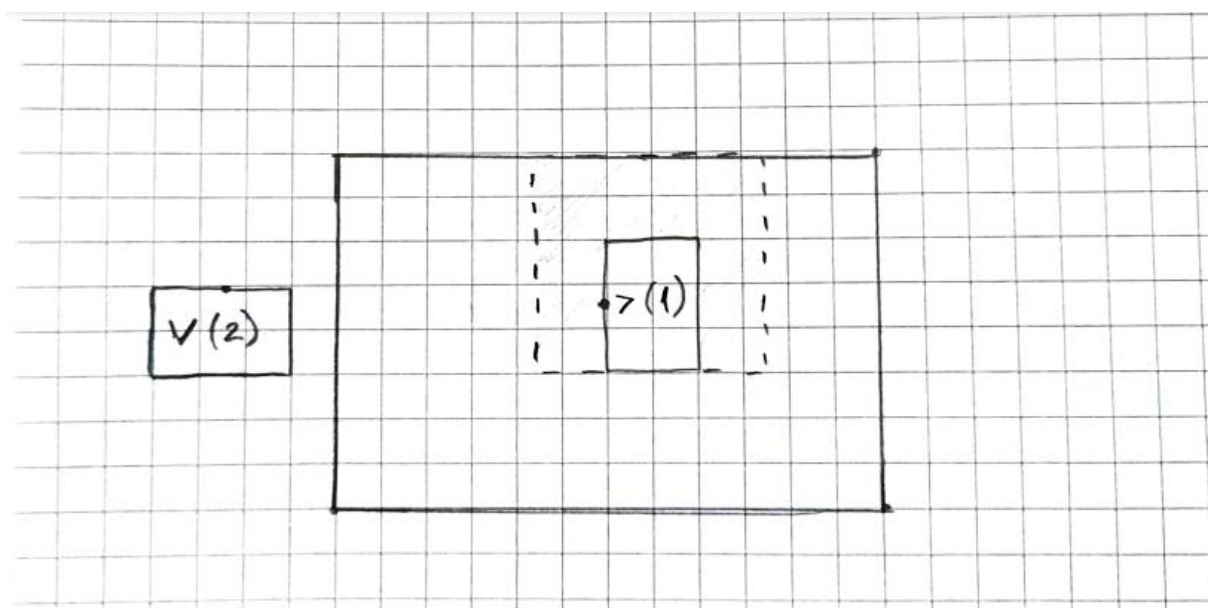


Figur 5.25. Exempel 2 på hur lådor inte ska överlappas.

Om man roterar lådan som ska placeras och låter den placerade lådans vinkel vara 0 grader så blir området enligt följande:

$$L1.posY - L2.längd/2 < L2.posY < L1.posY + L1.bredd + L2.längd/2$$

$$L1.posX - L1.längd/2 + L2.bredd < L2.posX < L1.posX + L1.längd/2$$



Figur 5.26. Exempel 3 på hur lådor inte ska överlappas.

Området där nya lådor inte får placeras beror alltså både på vinkeln på den redan placerade lådan samt vinkeln på lådan som ska placeras. Det går alltså inte att skapa ett generellt område runt de placerade lådorna där ingen annan låda får placeras. Därför måste de bådvas vinklar jämföras innan området bestäms.

När alla lådor befinner sig på en plats som är godkänd kan man gå vidare och upprepa föregående steg för staplingsmönster B, om man valt att använda sig utav mer än ett

staplingsmönster. Om man bara valt att använda ett staplingsmönster så sparas receptet i stället för att fortsätta till nästa steg.

```
public partial class rSkapaSMApos
{
    void rSkapaSMApos_Opened(System.Object sender, System.EventArgs e)
    {
        if(Globals.Tags.V_AntallLavmönster.Value == 1)
        {
            this.btn_fortsätt.Visible = false;

            Globals.Tags.B_StaplingsmönsterLav1.Value = 0;
            Globals.Tags.B_StaplingsmönsterLav2.Value = 0;
            Globals.Tags.B_StaplingsmönsterLav3.Value = 0;
            Globals.Tags.B_StaplingsmönsterLav4.Value = 0;
            Globals.Tags.B_StaplingsmönsterLav5.Value = 0;
            Globals.Tags.B_StaplingsmönsterLav6.Value = 0;
            Globals.Tags.B_StaplingsmönsterLav7.Value = 0;
            Globals.Tags.B_StaplingsmönsterLav8.Value = 0;
            Globals.Tags.B_StaplingsmönsterLav9.Value = 0;
            Globals.Tags.B_StaplingsmönsterLav10.Value = 0;
            Globals.Tags.B_StaplingsmönsterLav11.Value = 0;
            Globals.Tags.B_StaplingsmönsterLav12.Value = 0;
            Globals.Tags.B_StaplingsmönsterLav13.Value = 0;
            Globals.Tags.B_StaplingsmönsterLav14.Value = 0;
            Globals.Tags.B_StaplingsmönsterLav15.Value = 0;
            Globals.Tags.B_StaplingsmönsterLav16.Value = 0;
        }
        else
        {
            this.btn_fortsätt.Visible = true;
        }
    }
}
```

*Figur 5.27. Kod som körs då man endast använder ett staplingsmönster*

Efter inställningarna för staplingsmönster B är klara så kommer man till skärmen där lavmönstret väljs. Genom att trycka på ett av laven så ändras det från A till B och sedan tillbaka när man trycker igen. För att hålla reda på vilket lav som ska ha vilket staplingsmönster så används följande register. Om en bit är en nolla så betyder det att det är staplingsmönster A som ska köras och om biten är en etta så är det staplingsmönster B.

B_StaplingsmönsterLav1	Bit	VAR_GLOBAL D2980.0
B_StaplingsmönsterLav2	Bit	VAR_GLOBAL D2980.1
B_StaplingsmönsterLav3	Bit	VAR_GLOBAL D2980.2
B_StaplingsmönsterLav4	Bit	VAR_GLOBAL D2980.3
B_StaplingsmönsterLav5	Bit	VAR_GLOBAL D2980.4
B_StaplingsmönsterLav6	Bit	VAR_GLOBAL D2980.5
B_StaplingsmönsterLav7	Bit	VAR_GLOBAL D2980.6
B_StaplingsmönsterLav8	Bit	VAR_GLOBAL D2980.7
B_StaplingsmönsterLav9	Bit	VAR_GLOBAL D2980.8
B_StaplingsmönsterLav10	Bit	VAR_GLOBAL D2980.9
B_StaplingsmönsterLav11	Bit	VAR_GLOBAL D2980.A
B_StaplingsmönsterLav12	Bit	VAR_GLOBAL D2980.B
B_StaplingsmönsterLav13	Bit	VAR_GLOBAL D2980.C
B_StaplingsmönsterLav14	Bit	VAR_GLOBAL D2980.D
B_StaplingsmönsterLav15	Bit	VAR_GLOBAL D2980.E
B_StaplingsmönsterLav16	Bit	VAR_GLOBAL D2980.F

Figur 5.28. Register för lavmönster.



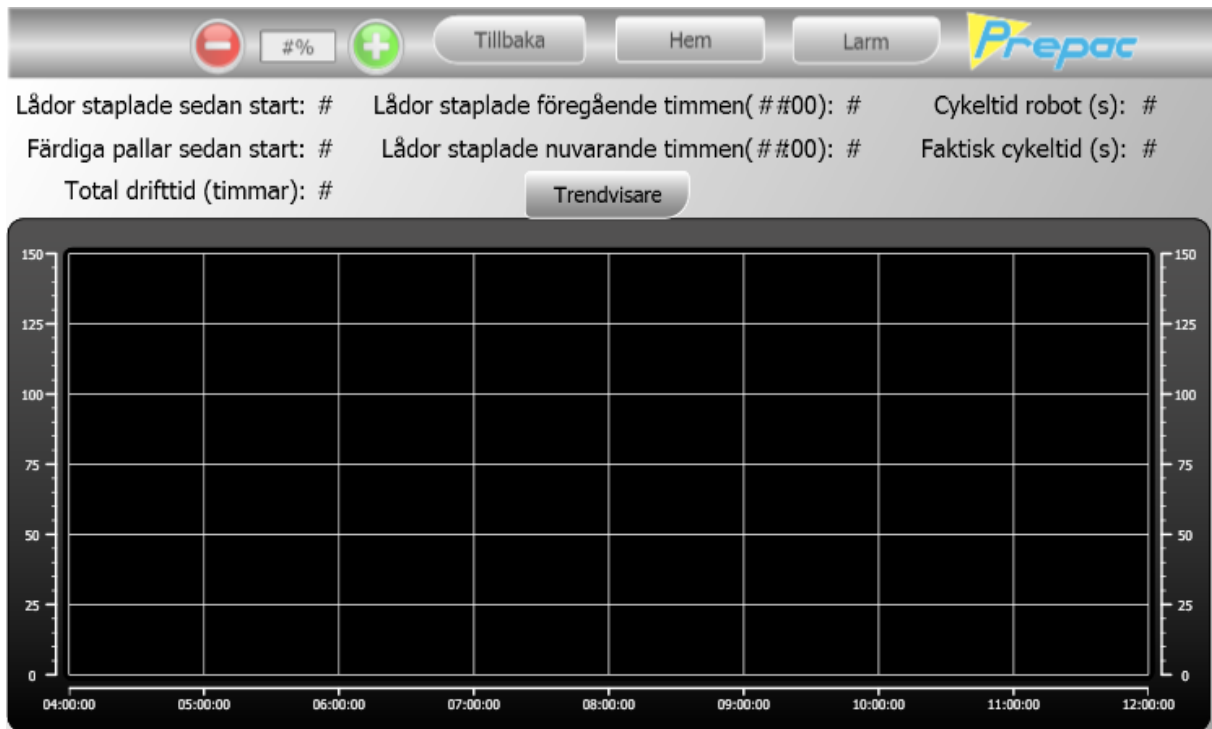
5.29. Val av lavmönster

När allt ovan är gjort så har alla taggarna som är med i receptet fått ett värde och man kan spara det som ett nytt recept eller spara över ett befintligt.






## 5.1.4 Statistik

På ”Statistik”-skärmen syns hur många lådor som staplats sedan maskinen startades, hur många som staplats nuvarande timme och hur många som staplades föregående timme. Man kan även se hur många pallar som blivit klara sedan maskinen startade, maskinens totala drifttid och robotens cykeltid.

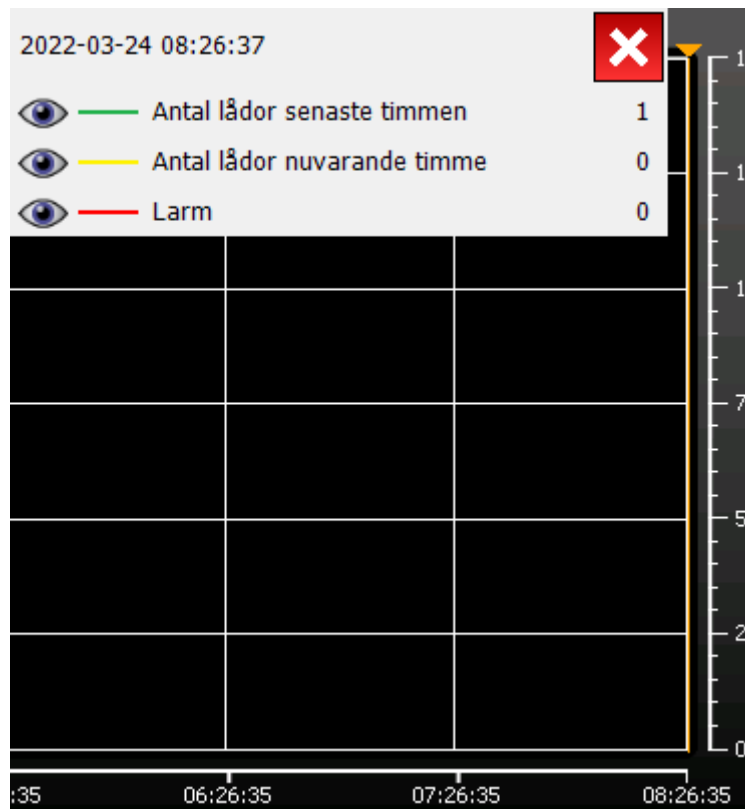


Figur 5.30. ”Statistik”-skärmen

I grafen så finns det tre kurvor. Det är lådor staplade nuvarande timme, föregående timme samt om där är något aktivt larm. Med hjälp av kurvorna kan man skapa sig en överblick på hur produktionen sett ut under dagen. Till exempel så kan man se om produktionen stått stilla ett tag och om det då var ett aktivt larm samtidigt kan det vara orsaken till stilleståndet. Man kan även gå tillbaka till en specifik tid i grafen med hjälp av trendvisaren. När man trycker på ”Trendvisare”-knappen så visas en pil längst upp på grafen som går att flytta fram och tillbaka. Det finns även möjlighet att dölja de olika kurvorna med denna funktion.

Name	Tag	Log Item	Expression	Color	Thickness
> Antal lådor se...	V_LådorSenas...			 34; 177; 7€	2
Antal lådor nu...	V_LådorNuvar...			 255; 242; 0	2
Larm	M_AktivtLarm...			 Red	2

Figur 5.31. Kurvor i grafen



Figur 5.32. Trendvisare

Koden som körs för att få fram statistik ser ut enligt följande. Först återställs alla taggarna när maskinen går ifrån avstängd till påslagen. Varje gång värdet på "D\_Aktuell\_Lådorstaplade" ändras så kommer lådor sedan start och lådor nuvarande timme att öka med ett. Om det nya värdet på "D\_Aktuell\_Lådorstaplade" skulle vara lika med "D\_AntalLåorPerPall" så har den sista lådan på pallen precis lagts och antalet pallar som blivit klara sedan start ökas med ett.

Taggen "V\_NuvarandeTimme" är en systemtagg som tar sitt värde direkt ifrån systemet, i detta fall vilken timme det är. Så när den taggen ändras betyder det att man gått över till en ny timme och då sker följande. Den föregående timmen är den nuvarande timmen minus ett. Lådor som staplades den föregående timmen är lika med så många lådor som staplats nuvarande timme och sen återställs taggen för lådor nuvarande timme.



```

void M_MaskinStartad_ValueChange(System.Object sender, Core.Api.DataSource.ValueChangedEventArgs e)//Reseta
{
    if(Globals.Tags.M_MaskinStartad.Value == 1)
    {
        Globals.Tags.V_LådorSedanStart.Value = 0;
        Globals.Tags.V_PallarSedanStart.Value = 0;
        Globals.Tags.V_LådorNuvarandeTimme.Value = 0;
        Globals.Tags.V_LådorSenasteTimmen.Value = 0;
    }
}

void D_Aktuell_LådorStaplade_ValueChange(System.Object sender, Core.Api.DataSource.ValueChangedEventArgs e)
{
    Globals.Tags.V_LådorSedanStart.Value = Globals.Tags.V_LådorSedanStart.Value + 1;
    Globals.Tags.V_LådorNuvarandeTimme.Value = Globals.Tags.V_LådorNuvarandeTimme.Value + 1;

    if(Globals.Tags.D_Aktuell_LådorStaplade.Value == Globals.Tags.D_AntalLådorPerPall.Value) //Ökar antalet
    {
        Globals.Tags.V_PallarSedanStart.Value = Globals.Tags.V_PallarSedanStart.Value + 1;
    }
}

void V_NuvarandeTimme_ValueChange(System.Object sender, Core.Api.DataSource.ValueChangedEventArgs e) //När
{
    Globals.Tags.V_FöregåendeTimme.Value = Globals.Tags.V_NuvarandeTimme.Value - 1; //Föregående timmen är
    Globals.Tags.V_LådorSenasteTimmen.Value = Globals.Tags.V_LådorNuvarandeTimme.Value; //Lådor staplade fö
    Globals.Tags.V_LådorNuvarandeTimme.Value = 0;
}

```

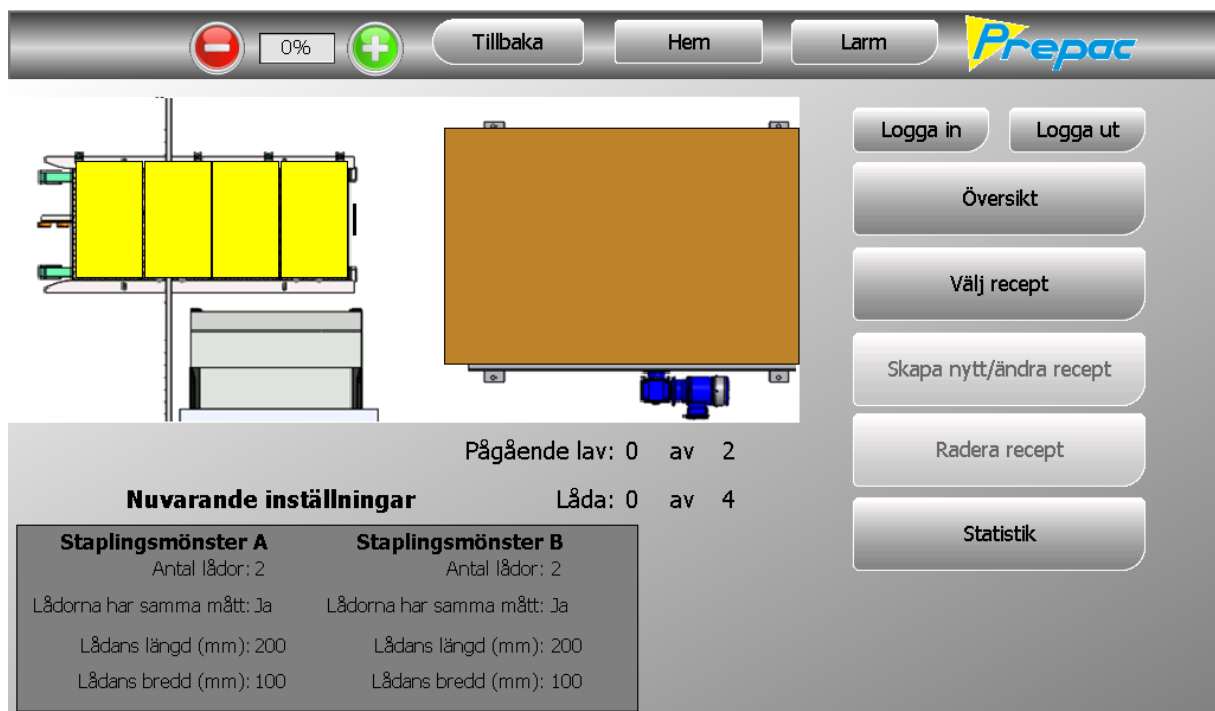
*Figur 5.33. Kod för att få fram önskad statistik*

## 5.1.5 Säkerhet

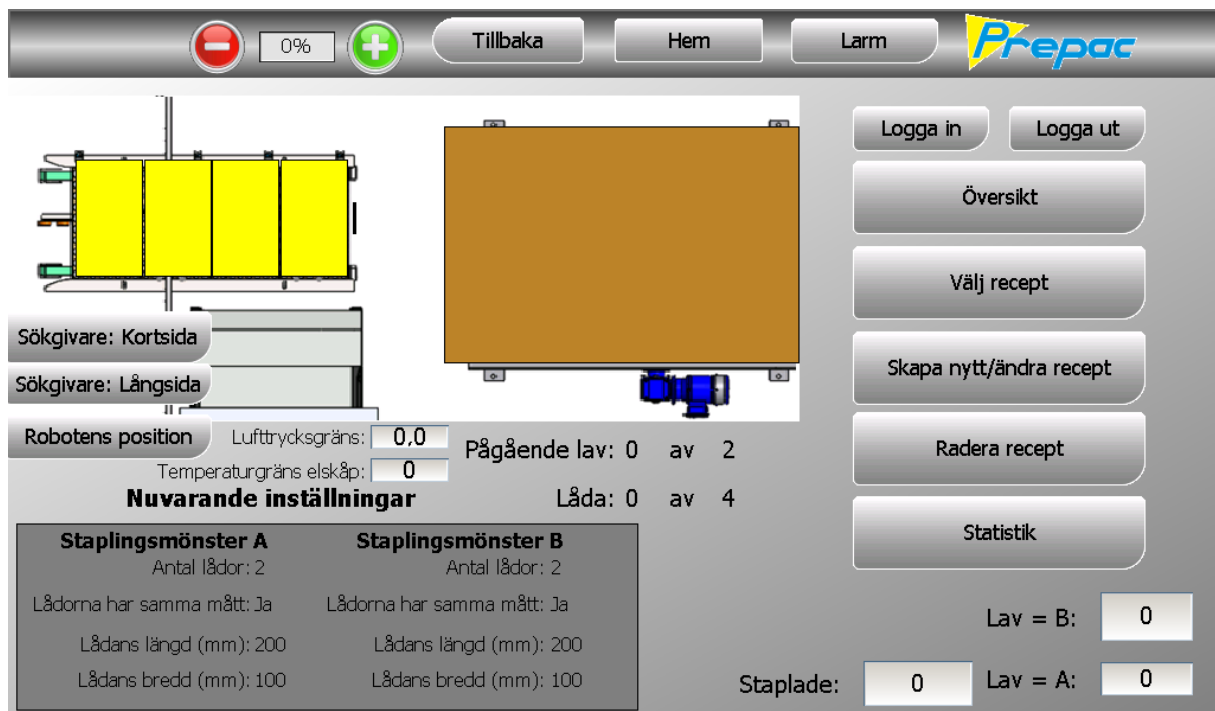
HMI-programmets säkerhet är baserat på ett RBAC-system (Role-Based Access Control). Man kan skapa användare i system och de tilldelas en roll eller grupp. Det går sedan att bestämma vilka rättigheter de olika grupperna har i systemet.

För att minimera risken att det sker några fel vid staplingen så kommer det krävas inloggning för att kunna ändra på några inställningar. Utan inloggning så kommer man endast kunna se översiktsskärmen, hemskärmen, se vilka de nuvarande inställningarna är samt ladda ett färdigt recept. Om man istället loggar in som administratör så får man tillgång till skapa/ändra och radera recept knapparna.

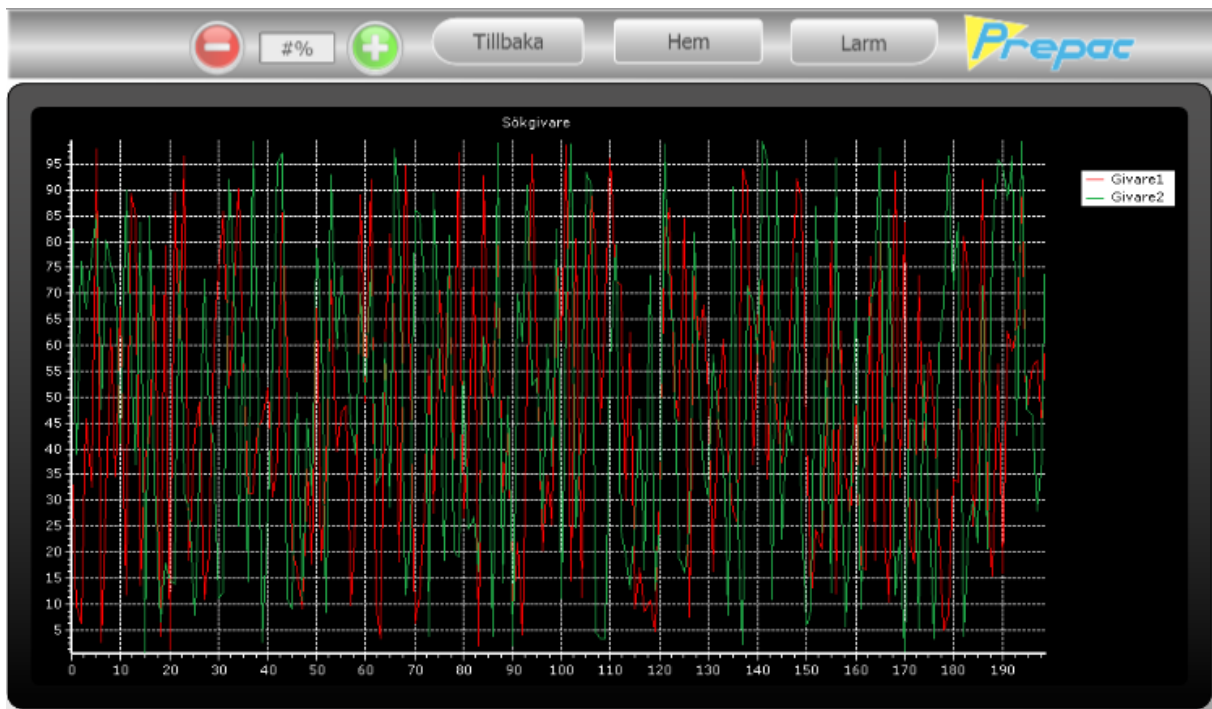
Det finns även möjlighet att logga in som skapare av programmet. Då får man tillgång till data som kan användas vid till exempel felsökning eller kalibrering av roboten men är ointressanta när den är i drift. Exempel på sådan data är grafer ifrån sökningar av pallen eller möjligheten att manuellt skicka positioner som roboten ska gå till. Som skapare kan man även ställa in gränserna för hur lågt lufttrycket och hur hög temperaturen får vara innan larmet går.



Figur 5.34. "Visualisering"-skärmen när man är utloggad



5.35. "Visualisering"-skärmen när man är inloggad som skapare

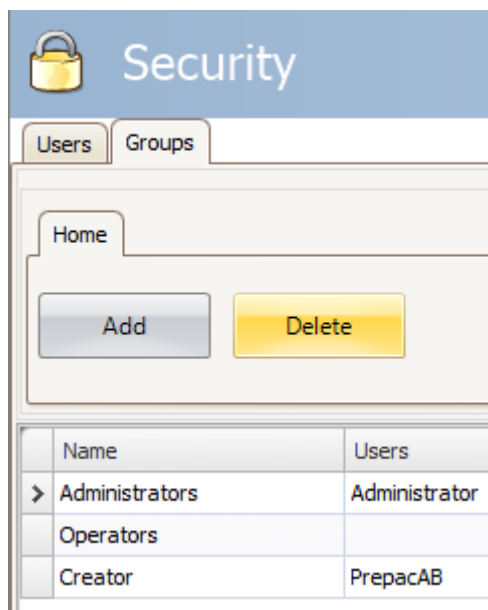


Figur 5.36. Sökning av pallens kortsida

Palletering		Nuvarande position	
Send pos X: #	Offset stapling X: #	RecPos Rob X: #	
Send pos Y: #	Offset stapling Y: #	RecPos Rob Y: #	
Send pos Z: #	Offset stapling Z: #	RecPos Rob Z: #	
Send pos RZ: #			
Pallhämtning		Lägen	
Send pos X: #	Offset hämtläge X: #	SendRob SökKortsida: #	
Send pos Y: #	Offset hämtläge Y: #	SendRob SökLångsida: #	
Send pos Z: #	Offset hämtläge Z: #	SendRob HämtaTråg: #	
Send pos RZ: #	Offset hämtläge RZ: #	SendRob HämtaPall: #	
Höjd efter 1a sök: #		Rob i hemposition: #	

Figur 5.37. Robotens position

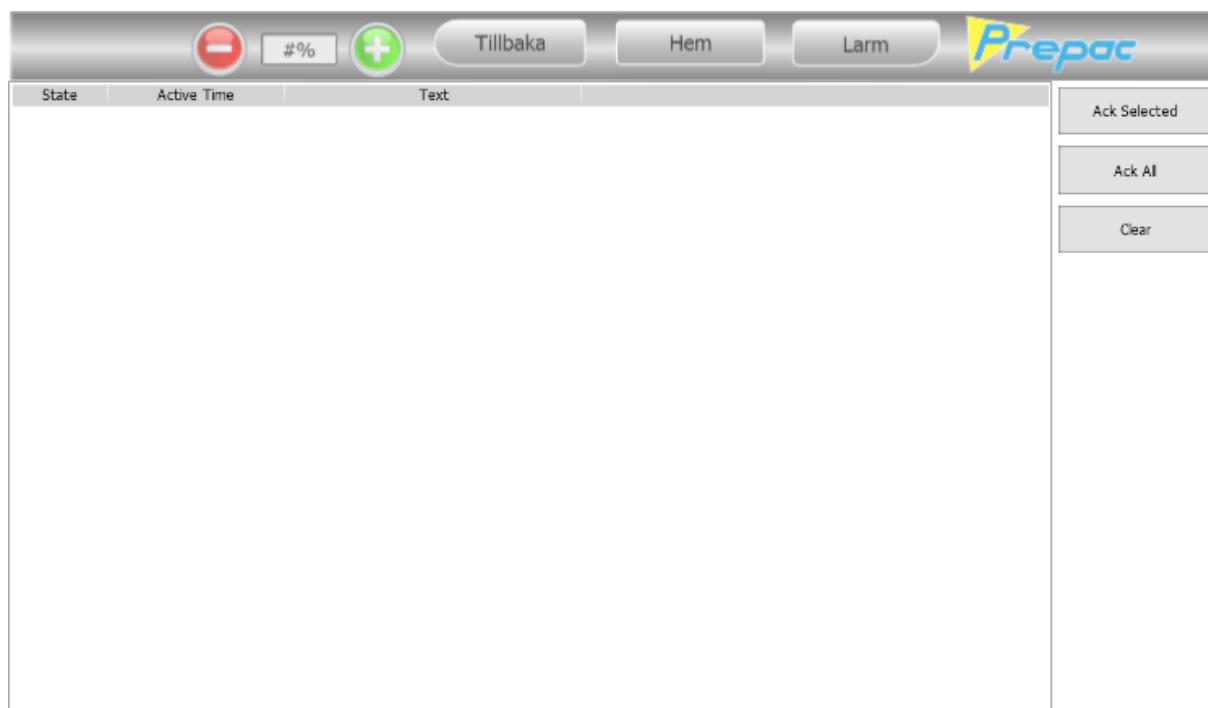
Inloggning sker med hjälp av logga in knappen på visualiseringsskärmen. Då väljer man vilken användare som ska logga in och matar sedan in lösenordet. Användaren är kopplad till en säkerhetsgrupp som i sin tur bestämmer om man får åtkomst till de olika funktionerna i programmet. För att logga ut trycker man bara på ”Logga ut”-knappen som ligger bredvid ”Logga in”-knappen på visualiseringsskärmen.



Figur 5.38. Säkerhetsgrupperna

## 5.1.6 Alarm

Alarm är viktiga för att meddela operatören att något är fel och måste åtgärdas.



Figur 5.39. "Larm"-skärmen

Name	Text	Tag	Expression	Condition	Trigger Value
SlutPåPallar	Fyll på pallar	M_LarmPallarSlut		EqualTo	1
LågtLufttryck	Lågt lufttryck	M_LarmLufttryckLågt		EqualTo	1
PallsökningMissly...	Pallsökning misslyckades	M_LarmPallsökningMisslyckad		EqualTo	1
TemperaturElskåp	Temperatur elskåpet	M_LarmTemperaturSkåp		EqualTo	1
FR.20	Frekvensomriktare 20	M_LarmFR.20		EqualTo	1
FR.30	Frekvensomriktare 30	M_LarmFR.30		EqualTo	1
Tråghämtning	Tråghämtning misslyckades	M_LarmTråghämtning		EqualTo	1
UtbonaFull	Utbanan är full	M_LarmUtbonaFull		EqualTo	1

Figur 5.40. Exempel på alarm

Det är PLCn som känner av när något är fel och ändrar på motsvarande tagg från 0 till 1. Då dyker en larmsymbol upp på panelen och när man går till larmskärmen så står det vad som är felet. Under tiden som ett larm är aktivt så kommer taggen "M\_AktivtLarmHMI" sättas till 1 och när larmet upphör så sätts den till 0.

The screenshot displays the configuration for an active alarm. On the left, a sidebar lists 'Actions' with sub-items: 'Alarm Distribution', 'General', and 'Server Mode'. The main configuration area shows two trigger configurations:

- Alarm active:** Triggered by 'Alarm active', it performs a 'Set Tag' action on 'M\_AktivtLarmHMI'.
- Alarm inactive:** Triggered by 'Alarm inactive', it performs a 'Reset Tag' action on 'M\_AktivtLarmHMI'.

Figur 5.41. Aktivt larm

## 6. Slutsats

Slutsatsen för hela examensarbetet kommer presenteras i följande kapitel. Problemen i problemformuleringen kommer att besvaras här. Dessutom kommer det göras en reflektion över etiska aspekter och vilka de framtida utvecklingsmöjligheterna är.

Det har skapats ett HMI-program till en robot som ska stapla lådor på en pall efter angivet staplingsmönster. Programmet innehåller en mängd funktioner så som recepthantering, alarm, accesskontroll, statistik och visualisering av staplingsprocessen.

Alla tänkbara funktioner som kan önskas av programmet har implementerats samt relevanta parametrar visas i programmet. För att minimera riskerna och felen vid körning så krävdes det inloggning för att få access till vissa funktioner så som radera recept och ändra larmgräns för lufttrycket. Lösningen för visualiseringen av staplings- och pallmönster gjordes på ett sådant sätt så att de är lätta att förstå vid första anblick. Programmet skulle även vara skalbart vilket det är inom rimliga gränser, mer om det under framtida utvecklingsmöjligheter.

### 6.1 Reflektion över etiska aspekter

Den andra punkten i ingenjörernas hederskodex lyder: ”Ingenjören bör sträva efter att förbättra tekniken och det tekniska kunnandet i riktning mot ett effektivare resursutnyttjande utan skadeverkningar”. Om inte roboten som ska stapla lådorna hade funnits så hade arbetet behövt göras av en människa. Dels så hade arbetet tagit längre tid och det hade inte utförts med samma precision, men det hade även funnits en skaderisk för de som arbetar då det är många repetitiva rörelser vilket till slut kunnat leda till skador. Alltså har arbetet genererat samhällsnytta genom att processen effektiviserats samt att eventuella slitageskador på människor har förhindrats.

### 6.2 Framtida utvecklingsmöjligheter

Det finns en del utvecklingsmöjligheter att genomföra i framtiden. Just nu kan HMI-programmet hantera 16 lav och två olika staplingsmönster. Det beror på att det är bitarna från en INT16 som används för att spara och ändra laven. För att utöka antalet lav och staplingsmönster så krävs det att man använder en INT16 per lav i stället för en Bit per lav. Det skulle båda göra att man inte är limiterad till 16 lav och varje lav är inte längre begränsad till att innehålla en etta eller nolla. Då varje lav kan innehålla 0, 1, 2, 3 ... osv går det lätt att översätta så att de motsvarar staplingsmönster A, B, C, D ... osv. Nackdelen är att det skulle ta upp fler platser i registret vilket man kanske inte alltid har tillgängligt, men vid behov så skulle det vara en fungerande lösning.

HMI-programmet klarar också bara 2 olika staplingsmönster för tillfället. Detta eftersom endast 2 staplingsmönster med 12 lådor i varje lagts in i registret. Det går att utöka genom att

börja staplingsmönster C och dess lådor på D5000 och staplingsmönster D på D6000. Rent teoretiskt skulle man kunna ha 100 lådor i varje lav i programmet, men i praktiken kommer det aldrig att hända. Skulle man dock vilja använda mer än 12 lådor per staplingsmönster så kräver det att man lägger till en skärm mellan figur 5.15 och 5.18 där man fortsätter att mata in bredd och längd på lådorna. Det kräver bara att man uppdaterar taggarna på textfälten och i koden för att det ska fungera.

I framtiden skulle man även kunna optimera koden som används genom att skapa klasser och metoder som körs. Det skulle göra den mycket mer läsbar och förståelig samt göra det enklare om några ändringar skulle behöva ske. Som det är nu så skulle det vara väldigt lätt att glömma genomföra en ändring på något ställe.

## 7. Terminologi

HMI	Human Machine Interface
PLC	Programmable Logic Controller
I/O-system	Input/Output-system
C#	Programmeringsspråk (C-sharp)
Staplingsmönster	I vilket mönster som man staplar lådorna i på ett lav.
Lavmönster	I vilken ordning staplingsmönsterna ska lavas på pallen.
Bit	Datatyp som kan anta värdet 0 eller 1.
INT16	Datatyp som består av 16 bitar.



## 8. Källförteckning

Feng, Lingyun och Xu, Yuehua. 2021. *Design and Application of HMI Systems of Robot Based on Vision*. Guangzhou, Kina: Guangdong Mechanical & Electrical Polytechnic.

<https://ieeexplore-ieee-org.ludwig.lub.lu.se/stamp/stamp.jsp?tp=&arnumber=9467535&fbclid=IwAR1rA7iLAIvgMs3c9FhdnOlaTonPetMnB-pWc2myhDCs5VNjL9YwtECxOBY&tag=1> (Hämtad: 2022-02-17) [1]

Elektroautomatik. *Vad är PLC?*. <https://elektroautomatik.se/automationsguider/vad-ar-plc/>. (Hämtad 2022-04-25) [2]

Beijer Electronics. *Advanced functionality*.

<https://www.beijerelectronics.com/en/Products/software/ix-hmi-software/for-advanced-users> (Hämtad 2022-02-17) [3]

Mitsubishi Electric. *GX Works3*.

[https://www.mitsubishielectric.com/fa/products/cnt/plceng/smerit/gx\\_works3/index.html](https://www.mitsubishielectric.com/fa/products/cnt/plceng/smerit/gx_works3/index.html) (Hämtad: 2022-03-30) [4]

ABB, *Robot studio*, <https://new.abb.com/products/robotics/sv/robotstudio> (Hämtad: 2022-03-04) [5]

Beijer Electronics. *X2 base 10 v2 HP*.

[https://www.beijerelectronics.com/en/Products/Operator\\_panels/X2\\_base/X2\\_base\\_v2\\_HP/X2\\_base\\_10\\_v2\\_HP](https://www.beijerelectronics.com/en/Products/Operator_panels/X2_base/X2_base_v2_HP/X2_base_10_v2_HP) (Hämtad: 2022-02-14) [6]

Mitsubishi Electric. *MELSEC IQ-F Series*.

<https://www.mitsubishielectric.com/fa/products/cnt/plcf/items/index.html> (Hämtad: 2022-03-04) [7]

ABB, *IRB 4600*, <https://new.abb.com/products/robotics/industrial-robots/irb-4600> (Hämtad: 2022-03-04) [8]

# 9 Appendix

Delar av kod från HMI-programmet.

```
1. //-----
2. // Press F1 to get help about using script.
3. // To access an object that is not located in the current class, start the call with
   Globals.
4. // When using events and timers be cautious not to generate memoryleaks,
5. // please see the help for more information.
6. //-----
7.
8. namespace Neo.ApplicationFramework.Generated
9. {
10.     using System.Windows.Forms;
11.     using System;
12.     using System.Drawing;
13.     using Neo.ApplicationFramework.Tools;
14.     using Neo.ApplicationFramework.Common.Graphics.Logic;
15.     using Neo.ApplicationFramework.Controls;
16.     using Neo.ApplicationFramework.Interfaces;
17.
18.
19.     public partial class Tags
20.     {
21.
22.         void V_Lufttrycksgräns_ValueChange(System.Object sender,
   Core.Api.DataSource.ValueChangedEventArgs e)//Sätter gräns för lufttryck till PLCn.
23.         {
24.             Globals.Tags.HMI_LarmgänsLufttryck.Value =
   Globals.Tags.V_Lufttrycksgräns.Value * 1000;
25.         }
26.
27.         void D_Lufttryck_ValueChange(System.Object sender,
   Core.Api.DataSource.ValueChangedEventArgs e)//Visar lufttryck i bar.
28.         {
29.             Globals.Tags.V_Lufttryck.Value = Globals.Tags.D_Lufttryck.Value /
   1000.000;
30.         }
31.
32.         void M_MaskinStartad_ValueChange(System.Object sender,
   Core.Api.DataSource.ValueChangedEventArgs e)//Resetar statistiktaggarna när man startar
   maskinen.
33.         {
34.             if(Globals.Tags.M_MaskinStartad.Value == 1)
35.             {
36.                 Globals.Tags.V_LådorSedanStart.Value = 0;
37.                 Globals.Tags.V_PallarSedanStart.Value = 0;
38.                 Globals.Tags.V_LådorNuvarandeTimme.Value = 0;
39.                 Globals.Tags.V_LådorSenasteTimmen.Value = 0;
40.             }
41.         }
42.
43.         void D_Aktuell_LådorStaplade_ValueChange(System.Object sender,
   Core.Api.DataSource.ValueChangedEventArgs e) //Ökar antalet staplade lådor när värdet för
   aktuella lådor staplade ändras.
44.         {
45.             Globals.Tags.V_LådorSedanStart.Value =
   Globals.Tags.V_LådorSedanStart.Value + 1;
46.             Globals.Tags.V_LådorNuvarandeTimme.Value =
   Globals.Tags.V_LådorNuvarandeTimme.Value + 1;
47.
48.             if(Globals.Tags.D_Aktuell_LådorStaplade.Value ==
   Globals.Tags.D_AntallLådorPerPall.Value) //Ökar antalet färdiga pallar när aktuella lådan
   är sista lådan på pallen.
49.             {
```

```

50.             Globals.Tags.V_PallarSedanStart.Value =
Globals.Tags.V_PallarSedanStart.Value + 1;
51.         }
52.     }
53.
54.     void V_NuvarandeTimme_ValueChange(System.Object sender,
Core.Api.DataSource.ValueChangedEventArgs e) //När det blir en ny timme körs koden nedan.
55.     {
56.         Globals.Tags.V_FöregåendeTimme.Value =
Globals.Tags.V_NuvarandeTimme.Value - 1; //Föregående timmen är nuvarande timme -1.
57.         Globals.Tags.V_LådorSenasteTimmen.Value =
Globals.Tags.V_LådorNuvarandeTimme.Value; //Lådor staplade föregående timme är antlet
lådor som staplats nuvarande timme (innan taggen resetas).
58.         Globals.Tags.V_LådorNuvarandeTimme.Value = 0;
59.     }
60.
61.     void D_Stat_CykeltidStapling_ValueChange(System.Object sender,
Core.Api.DataSource.ValueChangedEventArgs e)
62.     {
63.         Globals.Tags.V_CykeltidStapling.Value =
Globals.Tags.D_Stat_CykeltidStapling.Value / 10.0;
64.     }
65.
66.     void D_Stat_CykeltidFaktisk_ValueChange(System.Object sender,
Core.Api.DataSource.ValueChangedEventArgs e)
67.     {
68.         Globals.Tags.V_CykeltidFaktisk.Value =
Globals.Tags.D_Stat_CykeltidFaktisk.Value / 10.0;
69.     }
70.
71.
72.     }
73. }
74.

```

```

1. //-----
2. // Press F1 to get help about using script.
3. // To access an object that is not located in the current class, start the call with
Globals.
4. // When using events and timers be cautious not to generate memoryleaks,
5. // please see the help for more information.
6. //-----
7.
8. namespace Neo.ApplicationFramework.Generated
9. {
10.     using System.Windows.Forms;
11.     using System;
12.     using System.Drawing;
13.     using Neo.ApplicationFramework.Tools;
14.     using Neo.ApplicationFramework.Common.Graphics.Logic;
15.     using Neo.ApplicationFramework.Controls;
16.     using Neo.ApplicationFramework.Interfaces;
17.
18.
19.     public partial class rSkapaSMA
20.     {
21.
22.         void Button_Click(System.Object sender, System.EventArgs e) //När knappen
"Ok" trycks så körs följande kod.
23.         {
24.
25.             int nbr = Globals.Tags.D_AntalLådor_StaplingsmönsterA.Value;
//Värdet av taggen "D_AntalLådor_udda" läggs i variabeln "nbr".
26.
27.             if(nbr > 12)
28.             {
29.                 MessageBox.Show("Max 12 lådor/lav", "Info"); //Om man
försöker skriva in fler än 12 lådor så säger den ifrån.
30.             }

```

```

31.         else //Annars om nbr är mellan 0 och 12 så händer följande.
32.         {
33.             if(nbr > 0)
34.             {
35.                 this.txt10.Visible = true; //Om nbr är större än
0 så visas text och textfälten för låda 1.
36.                 this.txtLängd10.Visible = true;
37.                 this.txtBredd10.Visible = true;
38.                 this.txtfLängd10.Visible = true;
39.                 this.txtfBredd10.Visible = true;
40.             }
41.             else
42.             {
43.                 this.txt10.Visible = false; //Annars om nbr är
mindre än eller lika med 0 så visas de inte.
44.                 this.txtLängd10.Visible = false;
45.                 this.txtBredd10.Visible = false;
46.                 this.txtfLängd10.Visible = false;
47.                 this.txtfBredd10.Visible = false;
48.
49.                 Globals.Tags.D_P1_L1Längd.Value = 0; //Värdena
på längd och bredd sätts även till 0.
50.                 Globals.Tags.D_P1_L1Bredd.Value = 0;
51.                 //Detta görs för att rensa taggarna på gamla
värden om man går från ett större antal lådor till ett lägre antal.
52.             }
53.
54.             if(nbr > 1) //Samma kod upprepas för varje låda.
55.             {
56.                 this.txt20.Visible = true;
57.                 this.txtLängd20.Visible = true;
58.                 this.txtBredd20.Visible = true;
59.                 this.txtfLängd20.Visible = true;
60.                 this.txtfBredd20.Visible = true;
61.             }
62.             else
63.             {
64.                 this.txt20.Visible = false;
65.                 this.txtLängd20.Visible = false;
66.                 this.txtBredd20.Visible = false;
67.                 this.txtfLängd20.Visible = false;
68.                 this.txtfBredd20.Visible = false;
69.
70.                 Globals.Tags.D_P2_L1Längd.Value = 0;
71.                 Globals.Tags.D_P2_L1Bredd.Value = 0;
72.             }
73.
74.             if(nbr > 2)
75.             {
76.                 this.txt30.Visible = true;
77.                 this.txtLängd30.Visible = true;
78.                 this.txtBredd30.Visible = true;
79.                 this.txtfLängd30.Visible = true;
80.                 this.txtfBredd30.Visible = true;
81.             }
82.             else
83.             {
84.                 this.txt30.Visible = false;
85.                 this.txtLängd30.Visible = false;
86.                 this.txtBredd30.Visible = false;
87.                 this.txtfLängd30.Visible = false;
88.                 this.txtfBredd30.Visible = false;
89.
90.                 Globals.Tags.D_P3_L1Längd.Value = 0;
91.                 Globals.Tags.D_P3_L1Bredd.Value = 0;
92.             }
93.
94.             if(nbr > 3)
95.             {

```

```

96.         this.txt40.Visible = true;
97.         this.txtLängd40.Visible = true;
98.         this.txtBredd40.Visible = true;
99.         this.txtfLängd40.Visible = true;
100.            this.txtfBredd40.Visible = true;
101.         }
102.         else
103.         {
104.             this.txt40.Visible = false;
105.             this.txtLängd40.Visible = false;
106.             this.txtBredd40.Visible = false;
107.             this.txtfLängd40.Visible = false;
108.             this.txtfBredd40.Visible = false;
109.
110.
111.             Globals.Tags.D_P4_L1Längd.Value = 0;
112.             Globals.Tags.D_P4_L1Bredd.Value = 0;
113.         }
114.
115.         if(nbr > 4)
116.         {
117.             this.txt50.Visible = true;
118.             this.txtLängd50.Visible = true;
119.             this.txtBredd50.Visible = true;
120.             this.txtfLängd50.Visible = true;
121.             this.txtfBredd50.Visible = true;
122.         }
123.         else
124.         {
125.             this.txt50.Visible = false;
126.             this.txtLängd50.Visible = false;
127.             this.txtBredd50.Visible = false;
128.             this.txtfLängd50.Visible = false;
129.             this.txtfBredd50.Visible = false;
130.
131.             Globals.Tags.D_P5_L1Längd.Value = 0;
132.             Globals.Tags.D_P5_L1Bredd.Value = 0;
133.         }
134.
135.
136.         if(nbr > 5)
137.         {
138.             this.txt60.Visible = true;
139.             this.txtLängd60.Visible = true;
140.             this.txtBredd60.Visible = true;
141.             this.txtfLängd60.Visible = true;
142.             this.txtfBredd60.Visible = true;
143.         }
144.         else
145.         {
146.             this.txt60.Visible = false;
147.             this.txtLängd60.Visible = false;
148.             this.txtBredd60.Visible = false;
149.             this.txtfLängd60.Visible = false;
150.             this.txtfBredd60.Visible = false;
151.
152.             Globals.Tags.D_P6_L1Längd.Value = 0;
153.             Globals.Tags.D_P6_L1Bredd.Value = 0;
154.         }
155.
156.         if(nbr > 6)
157.         {
158.             this.txt70.Visible = true;
159.             this.txtLängd70.Visible = true;
160.             this.txtBredd70.Visible = true;
161.             this.txtfLängd70.Visible = true;
162.             this.txtfBredd70.Visible = true;
163.         }
164.         else
165.         {

```

```

166.         this.txt70.Visible = false;
167.         this.txtLängd70.Visible = false;
168.         this.txtBredd70.Visible = false;
169.         this.txtfLängd70.Visible = false;
170.         this.txtfBredd70.Visible = false;
171.
172.         Globals.Tags.D_P7_L1Längd.Value = 0;
173.         Globals.Tags.D_P7_L1Bredd.Value = 0;
174.     }
175.
176.     if(nbr > 7)
177.     {
178.         this.txt80.Visible = true;
179.         this.txtLängd80.Visible = true;
180.         this.txtBredd80.Visible = true;
181.         this.txtfLängd80.Visible = true;
182.         this.txtfBredd80.Visible = true;
183.     }
184.     else
185.     {
186.         this.txt80.Visible = false;
187.         this.txtLängd80.Visible = false;
188.         this.txtBredd80.Visible = false;
189.         this.txtfLängd80.Visible = false;
190.         this.txtfBredd80.Visible = false;
191.
192.         Globals.Tags.D_P8_L1Längd.Value = 0;
193.         Globals.Tags.D_P8_L1Bredd.Value = 0;
194.     }
195.
196.     if(nbr > 8)
197.     {
198.         this.txt90.Visible = true;
199.         this.txtLängd90.Visible = true;
200.         this.txtBredd90.Visible = true;
201.         this.txtfLängd90.Visible = true;
202.         this.txtfBredd90.Visible = true;
203.     }
204.     else
205.     {
206.         this.txt90.Visible = false;
207.         this.txtLängd90.Visible = false;
208.         this.txtBredd90.Visible = false;
209.         this.txtfLängd90.Visible = false;
210.         this.txtfBredd90.Visible = false;
211.
212.         Globals.Tags.D_P9_L1Längd.Value = 0;
213.         Globals.Tags.D_P9_L1Bredd.Value = 0;
214.     }
215.
216.     if(nbr > 9)
217.     {
218.         this.txt100.Visible = true;
219.         this.txtLängd100.Visible = true;
220.         this.txtBredd100.Visible = true;
221.         this.txtfLängd100.Visible = true;
222.         this.txtfBredd100.Visible = true;
223.     }
224.     else
225.     {
226.         this.txt100.Visible = false;
227.         this.txtLängd100.Visible = false;
228.         this.txtBredd100.Visible = false;
229.         this.txtfLängd100.Visible = false;
230.         this.txtfBredd100.Visible = false;
231.
232.         Globals.Tags.D_P10_L1Längd.Value = 0;
233.         Globals.Tags.D_P10_L1Bredd.Value = 0;
234.     }
235.

```

```

236.         if(nbr > 10)
237.         {
238.             this.txt110.Visible = true;
239.             this.txtLängd110.Visible = true;
240.             this.txtBredd110.Visible = true;
241.             this.txtfLängd110.Visible = true;
242.             this.txtfBredd110.Visible = true;
243.         }
244.         else
245.         {
246.             this.txt110.Visible = false;
247.             this.txtLängd110.Visible = false;
248.             this.txtBredd110.Visible = false;
249.             this.txtfLängd110.Visible = false;
250.             this.txtfBredd110.Visible = false;
251.
252.             Globals.Tags.D_P11_L1Längd.Value = 0;
253.             Globals.Tags.D_P11_L1Bredd.Value = 0;
254.         }
255.
256.         if(nbr > 11)
257.         {
258.             this.txt120.Visible = true;
259.             this.txtLängd120.Visible = true;
260.             this.txtBredd120.Visible = true;
261.             this.txtfLängd120.Visible = true;
262.             this.txtfBredd120.Visible = true;
263.         }
264.         else
265.         {
266.             this.txt120.Visible = false;
267.             this.txtLängd120.Visible = false;
268.             this.txtBredd120.Visible = false;
269.             this.txtfLängd120.Visible = false;
270.             this.txtfBredd120.Visible = false;
271.
272.             Globals.Tags.D_P12_L1Längd.Value = 0;
273.             Globals.Tags.D_P12_L1Bredd.Value = 0;
274.         }
275.     }
276. }
277.
278.     void btnALHSM_Click(System.Object sender, System.EventArgs e) //När
knappen "Alla lådor har samma värde" trycks så körs följande kod.
279.     {
280.
281.         int nbr =
Globals.Tags.D_AntallLådor_StaplingsmönsterA.Value; //"D_AntallLådor_udda"s värde läggs i
variablen "nbr".
282.
283.
284.         if(nbr > 1)
285.         {
286.             Globals.Tags.D_P2_L1Bredd.Value =
Globals.Tags.D_P1_L1Bredd.Value; //Om nbr är större än 1 så kommer taggarna för låda2s
längd och bredd att få samma värde som taggarna för låda1.
287.             Globals.Tags.D_P2_L1Längd.Value =
Globals.Tags.D_P1_L1Längd.Value;
288.         }
289.
290.         if(nbr > 2) //Samma kod upprepas för varje låda.
291.         {
292.             Globals.Tags.D_P3_L1Bredd.Value =
Globals.Tags.D_P1_L1Bredd.Value;
293.             Globals.Tags.D_P3_L1Längd.Value =
Globals.Tags.D_P1_L1Längd.Value;
294.         }
295.
296.         if(nbr > 3)
297.         {

```

```

298.                                     Globals.Tags.D_P4_L1Bredd.Value =
    Globals.Tags.D_P1_L1Bredd.Value;
299.                                     Globals.Tags.D_P4_L1Längd.Value =
    Globals.Tags.D_P1_L1Längd.Value;
300.                                     }
301.
302.                                     if(nbr > 4)
303.                                     {
304.                                     Globals.Tags.D_P5_L1Bredd.Value =
    Globals.Tags.D_P1_L1Bredd.Value;
305.                                     Globals.Tags.D_P5_L1Längd.Value =
    Globals.Tags.D_P1_L1Längd.Value;
306.                                     }
307.
308.                                     if(nbr > 5)
309.                                     {
310.                                     Globals.Tags.D_P6_L1Bredd.Value =
    Globals.Tags.D_P1_L1Bredd.Value;
311.                                     Globals.Tags.D_P6_L1Längd.Value =
    Globals.Tags.D_P1_L1Längd.Value;
312.                                     }
313.
314.                                     if(nbr > 6)
315.                                     {
316.                                     Globals.Tags.D_P7_L1Bredd.Value =
    Globals.Tags.D_P1_L1Bredd.Value;
317.                                     Globals.Tags.D_P7_L1Längd.Value =
    Globals.Tags.D_P1_L1Längd.Value;
318.                                     }
319.
320.                                     if(nbr > 7)
321.                                     {
322.                                     Globals.Tags.D_P8_L1Bredd.Value =
    Globals.Tags.D_P1_L1Bredd.Value;
323.                                     Globals.Tags.D_P8_L1Längd.Value =
    Globals.Tags.D_P1_L1Längd.Value;
324.                                     }
325.
326.                                     if(nbr > 8)
327.                                     {
328.                                     Globals.Tags.D_P9_L1Bredd.Value =
    Globals.Tags.D_P1_L1Bredd.Value;
329.                                     Globals.Tags.D_P9_L1Längd.Value =
    Globals.Tags.D_P1_L1Längd.Value;
330.                                     }
331.
332.                                     if(nbr > 9)
333.                                     {
334.                                     Globals.Tags.D_P10_L1Bredd.Value =
    Globals.Tags.D_P1_L1Bredd.Value;
335.                                     Globals.Tags.D_P10_L1Längd.Value =
    Globals.Tags.D_P1_L1Längd.Value;
336.                                     }
337.
338.                                     if(nbr > 10)
339.                                     {
340.                                     Globals.Tags.D_P11_L1Bredd.Value =
    Globals.Tags.D_P1_L1Bredd.Value;
341.                                     Globals.Tags.D_P11_L1Längd.Value =
    Globals.Tags.D_P1_L1Längd.Value;
342.                                     }
343.
344.                                     if(nbr > 11)
345.                                     {
346.                                     Globals.Tags.D_P12_L1Bredd.Value =
    Globals.Tags.D_P1_L1Bredd.Value;
347.                                     Globals.Tags.D_P12_L1Längd.Value =
    Globals.Tags.D_P1_L1Längd.Value;
348.                                     }
349.

```



```

350.         }
351.
352.         void rSkapaSMA_Opened(System.Object sender, System.EventArgs e)
//Följande kod körs när skärmen "rSkapaUdda" öppnas.
353.         {
354.             int nbr =
Globals.Tags.D_AntallLådor_StaplingsmönsterA.Value; //Detta är samma kod som när man
trycker på knappen "OK".
355.             //Den körs här igen för att den ska visa rätt antal lådor
och deras värde om man redan laddat in ett recept.
356.
357.             if(nbr > 0)
358.             {
359.                 this.txt10.Visible = true;
360.                 this.txtLängd10.Visible = true;
361.                 this.txtBredd10.Visible = true;
362.                 this.txtfLängd10.Visible = true;
363.                 this.txtfBredd10.Visible = true;
364.             }
365.             else
366.             {
367.                 this.txt10.Visible = false;
368.                 this.txtLängd10.Visible = false;
369.                 this.txtBredd10.Visible = false;
370.                 this.txtfLängd10.Visible = false;
371.                 this.txtfBredd10.Visible = false;
372.
373.                 Globals.Tags.D_P1_L1Längd.Value = 0;
374.                 Globals.Tags.D_P1_L1Bredd.Value = 0;
375.             }
376.
377.             if(nbr > 1)
378.             {
379.                 this.txt20.Visible = true;
380.                 this.txtLängd20.Visible = true;
381.                 this.txtBredd20.Visible = true;
382.                 this.txtfLängd20.Visible = true;
383.                 this.txtfBredd20.Visible = true;
384.             }
385.             else
386.             {
387.                 this.txt20.Visible = false;
388.                 this.txtLängd20.Visible = false;
389.                 this.txtBredd20.Visible = false;
390.                 this.txtfLängd20.Visible = false;
391.                 this.txtfBredd20.Visible = false;
392.
393.                 Globals.Tags.D_P2_L1Längd.Value = 0;
394.                 Globals.Tags.D_P2_L1Bredd.Value = 0;
395.             }
396.
397.             if(nbr > 2)
398.             {
399.                 this.txt30.Visible = true;
400.                 this.txtLängd30.Visible = true;
401.                 this.txtBredd30.Visible = true;
402.                 this.txtfLängd30.Visible = true;
403.                 this.txtfBredd30.Visible = true;
404.             }
405.             else
406.             {
407.                 this.txt30.Visible = false;
408.                 this.txtLängd30.Visible = false;
409.                 this.txtBredd30.Visible = false;
410.                 this.txtfLängd30.Visible = false;
411.                 this.txtfBredd30.Visible = false;
412.
413.                 Globals.Tags.D_P3_L1Längd.Value = 0;
414.                 Globals.Tags.D_P3_L1Bredd.Value = 0;
415.             }

```

```

416.
417.
418.
419.
420.
421.
422.
423.
424.
425.
426.
427.
428.
429.
430.
431.
432.
433.
434.
435.
436.
437.
438.
439.
440.
441.
442.
443.
444.
445.
446.
447.
448.
449.
450.
451.
452.
453.
454.
455.
456.
457.
458.
459.
460.
461.
462.
463.
464.
465.
466.
467.
468.
469.
470.
471.
472.
473.
474.
475.
476.
477.
478.
479.
480.
481.
482.
483.
484.
485.

if(nbr > 3)
{
    this.txt40.Visible = true;
    this.txtLängd40.Visible = true;
    this.txtBredd40.Visible = true;
    this.txtfLängd40.Visible = true;
    this.txtfBredd40.Visible = true;
}
else
{
    this.txt40.Visible = false;
    this.txtLängd40.Visible = false;
    this.txtBredd40.Visible = false;
    this.txtfLängd40.Visible = false;
    this.txtfBredd40.Visible = false;

    Globals.Tags.D_P4_L1Längd.Value = 0;
    Globals.Tags.D_P4_L1Bredd.Value = 0;
}

if(nbr > 4)
{
    this.txt50.Visible = true;
    this.txtLängd50.Visible = true;
    this.txtBredd50.Visible = true;
    this.txtfLängd50.Visible = true;
    this.txtfBredd50.Visible = true;
}
else
{
    this.txt50.Visible = false;
    this.txtLängd50.Visible = false;
    this.txtBredd50.Visible = false;
    this.txtfLängd50.Visible = false;
    this.txtfBredd50.Visible = false;

    Globals.Tags.D_P5_L1Längd.Value = 0;
    Globals.Tags.D_P5_L1Bredd.Value = 0;
}

if(nbr > 5)
{
    this.txt60.Visible = true;
    this.txtLängd60.Visible = true;
    this.txtBredd60.Visible = true;
    this.txtfLängd60.Visible = true;
    this.txtfBredd60.Visible = true;
}
else
{
    this.txt60.Visible = false;
    this.txtLängd60.Visible = false;
    this.txtBredd60.Visible = false;
    this.txtfLängd60.Visible = false;
    this.txtfBredd60.Visible = false;

    Globals.Tags.D_P6_L1Längd.Value = 0;
    Globals.Tags.D_P6_L1Bredd.Value = 0;
}

if(nbr > 6)
{
    this.txt70.Visible = true;
    this.txtLängd70.Visible = true;
    this.txtBredd70.Visible = true;
    this.txtfLängd70.Visible = true;
    this.txtfBredd70.Visible = true;
}
else

```

```

486.     {
487.         this.txt70.Visible = false;
488.         this.txtLängd70.Visible = false;
489.         this.txtBredd70.Visible = false;
490.         this.txtfLängd70.Visible = false;
491.         this.txtfBredd70.Visible = false;
492.
493.         Globals.Tags.D_P7_L1Längd.Value = 0;
494.         Globals.Tags.D_P7_L1Bredd.Value = 0;
495.     }
496.
497.     if(nbr > 7)
498.     {
499.         this.txt80.Visible = true;
500.         this.txtLängd80.Visible = true;
501.         this.txtBredd80.Visible = true;
502.         this.txtfLängd80.Visible = true;
503.         this.txtfBredd80.Visible = true;
504.     }
505.     else
506.     {
507.         this.txt80.Visible = false;
508.         this.txtLängd80.Visible = false;
509.         this.txtBredd80.Visible = false;
510.         this.txtfLängd80.Visible = false;
511.         this.txtfBredd80.Visible = false;
512.
513.         Globals.Tags.D_P8_L1Längd.Value = 0;
514.         Globals.Tags.D_P8_L1Bredd.Value = 0;
515.     }
516.
517.     if(nbr > 8)
518.     {
519.         this.txt90.Visible = true;
520.         this.txtLängd90.Visible = true;
521.         this.txtBredd90.Visible = true;
522.         this.txtfLängd90.Visible = true;
523.         this.txtfBredd90.Visible = true;
524.     }
525.     else
526.     {
527.         this.txt90.Visible = false;
528.         this.txtLängd90.Visible = false;
529.         this.txtBredd90.Visible = false;
530.         this.txtfLängd90.Visible = false;
531.         this.txtfBredd90.Visible = false;
532.
533.         Globals.Tags.D_P9_L1Längd.Value = 0;
534.         Globals.Tags.D_P9_L1Bredd.Value = 0;
535.     }
536.
537.     if(nbr > 9)
538.     {
539.         this.txt100.Visible = true;
540.         this.txtLängd100.Visible = true;
541.         this.txtBredd100.Visible = true;
542.         this.txtfLängd100.Visible = true;
543.         this.txtfBredd100.Visible = true;
544.     }
545.     else
546.     {
547.         this.txt100.Visible = false;
548.         this.txtLängd100.Visible = false;
549.         this.txtBredd100.Visible = false;
550.         this.txtfLängd100.Visible = false;
551.         this.txtfBredd100.Visible = false;
552.
553.         Globals.Tags.D_P10_L1Längd.Value = 0;
554.         Globals.Tags.D_P10_L1Bredd.Value = 0;
555.     }

```

```

556.
557.
558.         if(nbr > 10)
559.         {
560.             this.txt110.Visible = true;
561.             this.txtLängd110.Visible = true;
562.             this.txtBredd110.Visible = true;
563.             this.txtfLängd110.Visible = true;
564.             this.txtfBredd110.Visible = true;
565.         }
566.         else
567.         {
568.             this.txt110.Visible = false;
569.             this.txtLängd110.Visible = false;
570.             this.txtBredd110.Visible = false;
571.             this.txtfLängd110.Visible = false;
572.             this.txtfBredd110.Visible = false;
573.
574.             Globals.Tags.D_P11_L1Längd.Value = 0;
575.             Globals.Tags.D_P11_L1Bredd.Value = 0;
576.         }
577.         if(nbr > 11)
578.         {
579.             this.txt120.Visible = true;
580.             this.txtLängd120.Visible = true;
581.             this.txtBredd120.Visible = true;
582.             this.txtfLängd120.Visible = true;
583.             this.txtfBredd120.Visible = true;
584.         }
585.         else
586.         {
587.             this.txt120.Visible = false;
588.             this.txtLängd120.Visible = false;
589.             this.txtBredd120.Visible = false;
590.             this.txtfLängd120.Visible = false;
591.             this.txtfBredd120.Visible = false;
592.
593.             Globals.Tags.D_P12_L1Längd.Value = 0;
594.             Globals.Tags.D_P12_L1Bredd.Value = 0;
595.         }
596.     }
597. }
598. }

```