1 Requirement specification

1.1 System definition

The system is a computerized sorting system which sort out defective items during production of transparent blocks. It must relieve human operators of the defective item selection during operation. The system also controls the conveyer belt where the items are placed before sorting.

Drawing here!!

1.1.1 Requirements

1. The system shall sort out non-transparent blocks.
2. The system shall not be slower than a human operator performing the same task.
3. The system shall have a start/stop button to start and stop the conveyer belt.
4. When the conveyer belt is started the system shall be fully functional.
5. The system shall be implemented using the LEGO NXT processor.
6. The system shall be implemented using available sensors and actuators for the LEGO NXT processor.
7. A demonstration prototype must be realizable within a short time period.
8. The system shall be implemented using COTS items to the greatest extent possible.

1.1.2 Problem Domain Structure

1.1.2.1 The 3-layered model of the system

1.1.3 Application Domain Structure

1.1.4 Acceptance Test Specification

2 Implementation

2.1 Investigation

2.1.1 The Touch sensor

The Touch sensor is used to start and stop the conveyer belt, and is the simplest type of the sensor available.

It will, when activated, allow or prevent electrical current to flow through it when it's activated; a normally open activation sensor or switch will allow current when activated and prevent current when not activated. A normally closed switch has the opposite function.

A simple monitoring strategy is to periodically poll or sample the switch's state; on / off.

Two things need to be considered when polling a switch:

- The polling frequency needs to be determined so the system has an acceptable responsiveness.
- A strategy for handling noise/prell on the switch output.

The handling of prell can be implemented in software, by having a relatively high sampling rate and only consider a change in the switch's state, when having detected the same value for e.g. 5-10 samples.

Another approach is to use dedicated HW to handle the prell, e.g. by incorporate a Schmitt-trigger circuit in the switch.

The trade off here is the need for higher sampling rates vs. a higher price for the switch.

2.1.1.1 Analysis

The LEGO Touch sensor comes with a Schmitt-trigger circuit; hence the software needn't consider prell.

Using the lejOS it's possible to setup an event listener to get the sensors state as a call-back in the application without using polling from the application.

The analysis consists of a small program which shall determine if the lejOS supported event listener is sufficient to get good responsiveness or if it is necessary to implement a polling scheme from the application.

2.1.1.2 Program

2.1.1.2.1 EntryPoint Class (Main)

```
package gis.group3.touchsensor.analysis;
import lejos.nxt.Button;
import lejos.nxt.SensorPort;
import lejos.nxt.TouchSensor;

public class EntryPoint {
	public static void main(String[] args) throws InterruptedException {
		TouchSensor ts = new TouchSensor(SensorPort.S1);
		SensorPort.S1.addSensorPortListener(new TouchSensorListener(ts));
		Button.ESCAPE.waitForPressAndRelease();
	}
}
```

2.1.1.2.2 TouchSensorListener Class

```
package gis.group3.touchsensor.analysis;
import lejos.nxt.LCD;
import lejos.nxt.SensorPort;
import lejos.nxt.SensorPortListener;
import lejos.nxt.TouchSensor;

public class TouchSensorListener implements SensorPortListener
{
	public TouchSensorListener(TouchSensor ts)
	{
		mts = ts;
	}
	public void stateChanged(SensorPort port, int oldValue, int newValue){

		if (mts.isPressed())
		{
			LCD.drawString("Touch pressed", 1, 2);
		}
```

```
                else
                {
                        LCD.clear();
                }
                LCD.refresh();
        }
        TouchSensor mts;
}
```

2.1.1.3 Conclusion

The program shows that the lejOS support for Touch sensor using event listeners is in fact highly responsive and the conclusion is therefore that the event listener can be used for the application.

2.1.2 The conveyer belt motor

A simple motor can be used for the belt drive since there are no requirements specifying that the belt must have adjustable speed. An A/C motor would be a simple and inexpensive choice for the application.

However, for the prototype the standard LEGO Servo motor is selected. The Servo motor is a stepper motor which is a flexible and widely used motor in many applications.

2.1.2.1 Analysis

As described, there are no requirements specifying that the belt must have adjustable speed; hence the motor is simply used at maximum speed. If a speed control were to be added, the power vs. speed ratio should be investigated, using the motors build-in taco meter.

2.1.2.2 Conclusion

The motor is used at maximum speed since no speed control is needed. In order to control the motor, using the Touch sensor, the previous program used for analysis of the Touch sensor is changed to start/stop the motor instead of writing to the LCD. However the Touch sensor shall be used as a toggle switch; the first activation starts the motor, the second stops the motor.