UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

TITLE OF THE THESIS

A Master's Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Rafel Mormeneo Melich

In partial fulfillment

of the requirements for the degree of

MASTER IN TELECOMMUNICATIONS ENGINEERING

Advisor: Javier Ruiz Hidalgo

Barcelona, June 2015

Title of the thesis: ......

Author: Rafel Mormeneo Melich

Advisor: Javier Ruiz Hidalgo

Abstract

Every copy of the thesis must include an abstract. The abstract should provide a concise summary of the thesis and should be a miniature version of the thesis in style, consisting of the following: a brief introduction, a summary of the results and the conclusions or main arguments presented in the thesis. The abstract for a Master's thesis may not exceed 150 words.

Dedication: A Dedication page may be included in your thesis just before the Acknowledgments page, but it is not a requirement.

Acknowledgements

It is advisable, but not mandatory, to declare the extent to which assistance has been given by members of the staff, fellow students, technicians or others in the gathering of materials and data; the design and construction of apparatus; the performance of experiments; the analysis of data, and the preparation of the thesis (including editorial help). In addition, the supervision and advice given by your advisor should also be acknowledged.

Revision history and approval record

Revision	Date	Purpose
0	23/03/2015	Document creation
1	06/05/2015	Preliminary Revision

Written by:		Reviewed and approved by:	
Date	dd/mm/yyyy	Date	dd/mm/yyyy
Name	Rafel Mormeneo Melich	Name	Javier Ruiz Hidalgo
Position	Project Author	Position	Project Supervisor

Table of contents

List of Figures

List of Tables

1. Introduction

An Introduction that clearly states the rationale of the thesis and including:

a. Requirements and specifications.

b. Statement of purpose (objectives).

c. Methods and procedures, citing if this work is a continuation of another project or it uses applications, algorithms, software or hardware previously developed by other authors.

d. Work plan with tasks, milestones and a Gantt diagram.

e. Description of the deviations from the initial plan and any eventualities that may have arisen.

The minimum number of chapters that this thesis document should contain are described below, although they may have different titles and more chapters may be added.

1.1. Summary

1.2. Field of application

This project relates to an industrial monitoring system. More specifically the main application of the developed device is to monitor, in real time, the position of a point machine.

A point machine is an electric motor driven switch that enables an operator to switch a train from one railroad track to another. The main elements of a switch are the points. Figure 1 shows the points in the normal position (top) and in the reverse position (bottom). When the points are in normal position the train, which goes from left to right in this diagram, continues by the same track. In the other hand, when the points are in reverse position the train changes from one track to another.

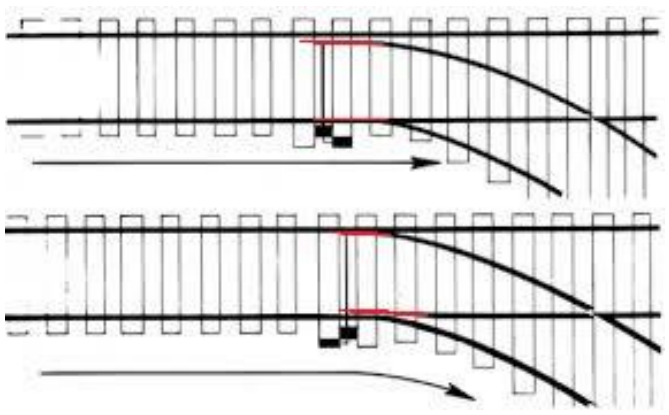


Figure 1. Points in a railroad switch. Top: Normal position; Bottom: Reverse position

Nowadays point machines are typically operated from a remote location. Because their closure is imperative it has a device to inform the operator about its current position. The most common realization of this device consists of two locking bars or detector bars, as we can see in Figure 2. The motor is connected to the stretcher (or throw bar) through gears. At the other end of the bar there are two points, or switch rails, attached to it. Each point has a locking bar attached to it. These locking bars go from the points to the engine closing. In the engine side there are two holding elements, called hammers, which are used to lock the locking bars. Each locking bar has a notch that allows the hammer to lock it in the current position. When the points are in normal position the first hammer locks one bar and when the points are in reverse position the second hammer locks the other bar.

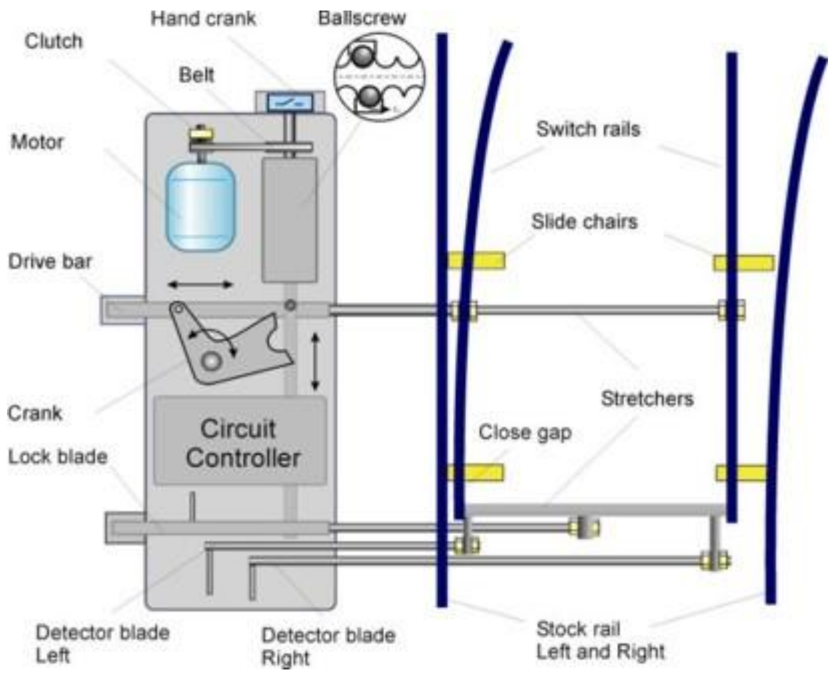


Figure 2. Parts of a point machine

Figure 3 shows a real point machine engine. We can see locking hammers are one in front of the other and the locking bars are one above the other. Figure 4 shows a detail of the hammers and the notch in the locking bars. Images are taken at different time instants and at different positions of the point machine. In the image on the left the hammer on the top fits the notch of one locking bar. In the other position, the hammer in the bottom fits the notch of the other bar.

When the operator moves the motor causes the linear and perpendicular movement of the drive bar. In turn, this causes the points to move and change their position. The points drag the locking bars which they are attached to. Inside of the engine housing, the hammer locks the locking bar corresponding to the side where the points are. This produces an electric signal that informs the remote operator that the switch has been successfully completed. When the locking bar does not arrive to its final position the electric circuit remains opened so the position of the switch is unknown and the operator cannot operate the switch. Security rules establish that a train cannot pass through an intersection where the point machine is in an unknown position. This affects directly the railway traffic in a high demand network like a subway or the suburban train.

Although the position of the point machine is the only information required from the point of view of railway safety it is not enough from the point of view of the maintenance of a big number of point machines. There exists devices like the one described in [1] which uses inductive proximity sensors or [2] which uses a transformer with two coils to detect the position of the bar. There exists another device developed in the framework of a master thesis at UPC that monitors all the signals available in the point machine including the exact position of the lock bars. This device is part of a system that will be further explained in Chapter 2.

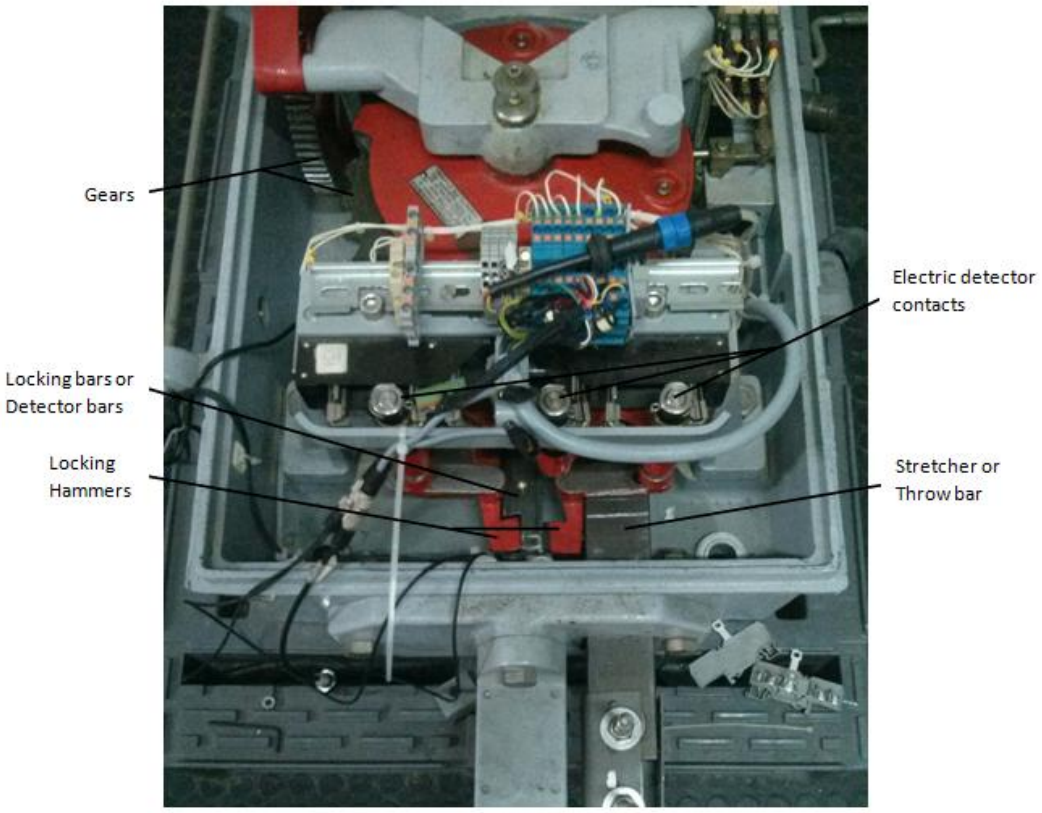


Figure 3. Parts of a real point machine

1.3. Goal

The goal of the present project is to develop a device to monitor the exact position of the lock bars. The task of the monitoring device can be divided in two sub-tasks. The first one consists in detecting the position of the point machine, this is, which lock bar is currently locked. The second one consists on measuring the real gap between the lock bar notch and the lock blade as we can see in Figure 4.

There are some requisites that this device has to accomplish.

1. Easiness to install. In the railway sector there is very limited period of time to do the maintenance tasks during the night where there is no operation of the service, therefore it is very important that the developed device to be very easy and quick to install.

2. Robustness. Sometimes point machines are in the exterior subject to bad weather, dust and humidity. Although the device will be installed inside of the motor housing it has to be robust to this unfavorable conditions.
3. Precision. It is very important that the developed solution gives exact measurements with high repetitiveness. The gap to be measured in the lock bar is of few millimeters, so the desired precision of the measurements is between 0.1mm and 0.5mm.
4. Reduced execution time. As it will be explained in Chapter 0 there are two modes of operation. In normal operation there is a window of 10 seconds to give the measure. In continuous operation the device has to be able to give at least one measure per second.
5. Reduced manufacturing costs. In a railway infrastructure are hundreds of engines to monitor, therefore, the unit price of the monitoring device must be low. The system should reduce the cost of infrastructure maintenance. Potential customers of the system are public or semi-public companies so the budget is considered at most for four years. All investments must have its pay-back less than this period of time.

We will have these points in mind in order to do the design of the device in Chapter 0.

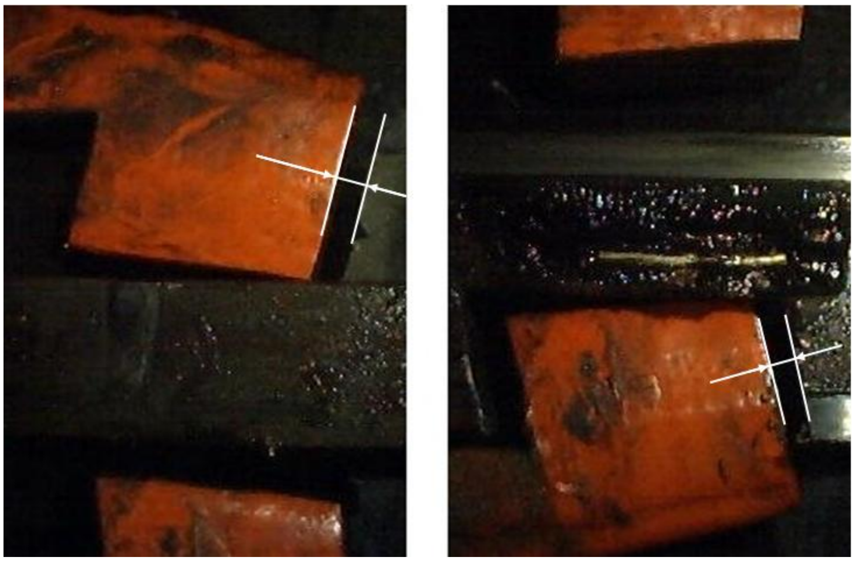


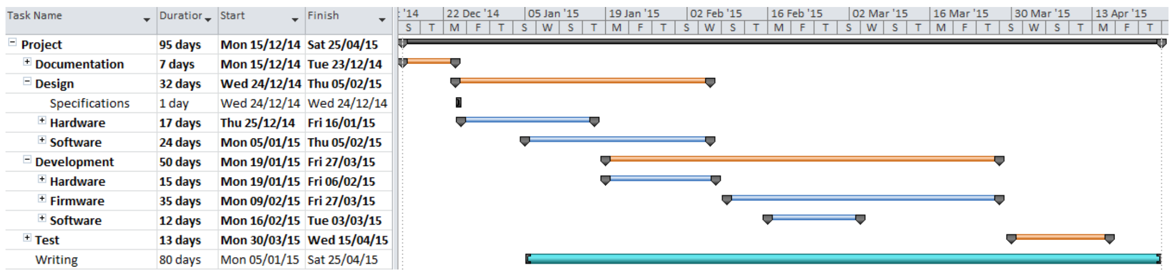
Figure 4. Gap to be measured. Left: Lower lock bar. Right: Upper lock bar

1.4. Planning

Figure 5 shows a summary of the Gantt Chart of the project and Figure 6 shows a more detailed planning of the phases and tasks. There are 5 people in the design and development team. This has been taking into account to establish the duration and planning of the tasks.

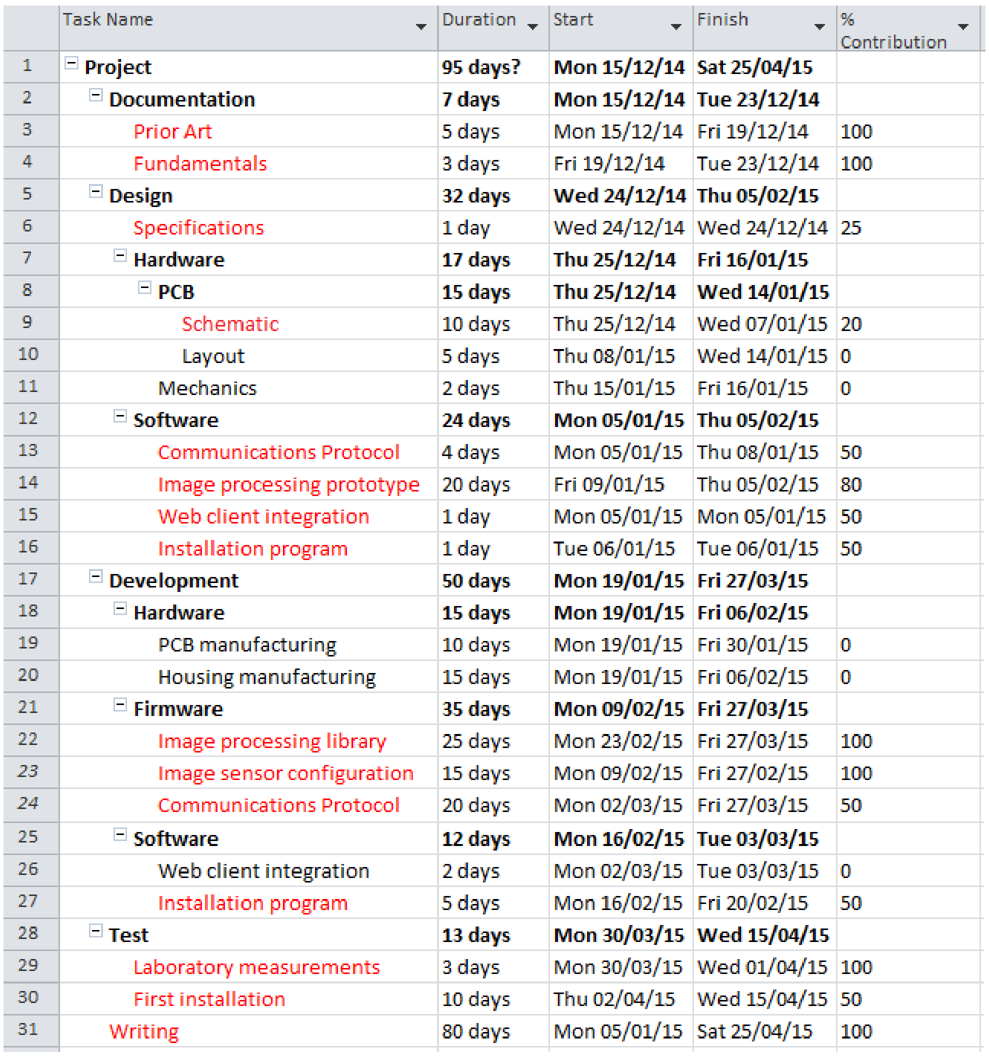
Projects where both, hardware and software, are involved are very hard to plan because sometimes there are several iterations in the hardware design. At the beginning of this project there is a first prototype with a basic functionality that consists on taking pictures and sending them to the Server. I have been involved in the hardware design, schematic and layout, and also in the firmware development of this prototype. Because of this, the planning of this project is more realistic and the hardware design is shorter than in other projects.

All members of the team are multidisciplinary but mainly the hardware team includes 3 people and the software and firmware team consists of 2 people. My tasks are mainly in the software and firmware development although as I have previously explained I have also been involved in the hardware development of the prototype which will be taken as the starting point of the final device. In order to make it more clear, I have marked in red the tasks where I have been actively involved in Figure 6.



Task Name	Duration	Start	Finish
Project	**95 days**	**Mon 15/12/14**	**Sat 25/04/15**
Documentation	**7 days**	**Mon 15/12/14**	**Tue 23/12/14**
Design	**32 days**	**Wed 24/12/14**	**Thu 05/02/15**
Specifications	1 day	Wed 24/12/14	Wed 24/12/14
Hardware	**17 days**	**Thu 25/12/14**	**Fri 16/01/15**
Software	**24 days**	**Mon 05/01/15**	**Thu 05/02/15**
Development	**50 days**	**Mon 19/01/15**	**Fri 27/03/15**
Hardware	**15 days**	**Mon 19/01/15**	**Fri 06/02/15**
Firmware	**35 days**	**Mon 09/02/15**	**Fri 27/03/15**
Software	**12 days**	**Mon 16/02/15**	**Tue 03/03/15**
Test	**13 days**	**Mon 30/03/15**	**Wed 15/04/15**
Writing	80 days	Mon 05/01/15	Sat 25/04/15

Figure 5. Planning. Gantt Chart Summary



	Task Name	Duration	Start	Finish	% Contribution
1	**Project**	**95 days?**	**Mon 15/12/14**	**Sat 25/04/15**	
2	**Documentation**	**7 days**	**Mon 15/12/14**	**Tue 23/12/14**	
3	Prior Art	5 days	Mon 15/12/14	Fri 19/12/14	100
4	Fundamentals	3 days	Fri 19/12/14	Tue 23/12/14	100
5	**Design**	**32 days**	**Wed 24/12/14**	**Thu 05/02/15**	
6	Specifications	1 day	Wed 24/12/14	Wed 24/12/14	25
7	**Hardware**	**17 days**	**Thu 25/12/14**	**Fri 16/01/15**	
8	**PCB**	**15 days**	**Thu 25/12/14**	**Wed 14/01/15**	
9	Schematic	10 days	Thu 25/12/14	Wed 07/01/15	20
10	Layout	5 days	Thu 08/01/15	Wed 14/01/15	0
11	Mechanics	2 days	Thu 15/01/15	Fri 16/01/15	0
12	**Software**	**24 days**	**Mon 05/01/15**	**Thu 05/02/15**	
13	Communications Protocol	4 days	Mon 05/01/15	Thu 08/01/15	50
14	Image processing prototype	20 days	Fri 09/01/15	Thu 05/02/15	80
15	Web client integration	1 day	Mon 05/01/15	Mon 05/01/15	50
16	Installation program	1 day	Tue 06/01/15	Tue 06/01/15	50
17	**Development**	**50 days**	**Mon 19/01/15**	**Fri 27/03/15**	
18	**Hardware**	**15 days**	**Mon 19/01/15**	**Fri 06/02/15**	
19	PCB manufacturing	10 days	Mon 19/01/15	Fri 30/01/15	0
20	Housing manufacturing	15 days	Mon 19/01/15	Fri 06/02/15	0
21	**Firmware**	**35 days**	**Mon 09/02/15**	**Fri 27/03/15**	
22	Image processing library	25 days	Mon 23/02/15	Fri 27/03/15	100
23	Image sensor configuration	15 days	Mon 09/02/15	Fri 27/02/15	100
24	Communications Protocol	20 days	Mon 02/03/15	Fri 27/03/15	50
25	**Software**	**12 days**	**Mon 16/02/15**	**Tue 03/03/15**	
26	Web client integration	2 days	Mon 02/03/15	Tue 03/03/15	0
27	Installation program	5 days	Mon 16/02/15	Fri 20/02/15	50
28	**Test**	**13 days**	**Mon 30/03/15**	**Wed 15/04/15**	
29	Laboratory measurements	3 days	Mon 30/03/15	Wed 01/04/15	100
30	First installation	10 days	Thu 02/04/15	Wed 15/04/15	50
31	Writing	80 days	Mon 05/01/15	Sat 25/04/15	100

Figure 6. Task Detail. The tasks where I have been actively involved are marked in red

2. State of the art of the technology used or applied in this thesis

2.1. Computer vision in industrial applications

Computer vision is becoming widely used in industry and many applications have appeared in recent years because it enables the automation of a wide range of processes.

Industrial computer vision applications can be classified in two groups. The first and most extended one consists on using computer vision to visual inspection. We can find many examples in the literature that uses image processing for this purpose. In [4] a method for locating and inspecting integrated circuit chips is described. Another application consists on automatic verification of the quality of printed circuit boards like in [5], fabric quality [7] or industrial plastic components [6].

The second group comprise the control of robots by artificial vision systems. One example of this application is robot guidance to a precise position or trajectory planning like in [8] and obstacle avoidance [9]. In [10] a more sophisticated method of an industrial application with two CCD sensors for spray painting of a general three dimensional surface is described.

In [11] a measurement technique using computer vision is described. We will explain a little more this article because it relates more directly to the task that we want to perform in this project. The goal of the work presented in this article is to measure area of leafs. In order to do so, the first step consists on obtaining binary images performing a segmentation of color images with an Otsu approach using the hue information. Otsu gives the optimal gray level in the range of [0,255] for image binarization. This will be further explained in Chapter 3. Due to some colors in background are close to the color of the leafs noise appears in the binary image. An opening is applied in the binary image to delete this kind of noise. Once they have filtered binary images they look for connected components using a two-scan algorithm. After the labeling step they use geometric characteristics of the a leaf in order to filter residual noise. Finally they count the foreground pixels and multiply this number by a precomputed constant to compute the leaf area. This constant is initialized with a calibration card of a known area. Similarly in this project we will follow more or less the same strategy to determine the exact position of the lock bar as it will be explained in Chapter 0.

2.2. Embedded computer vision

Because of image processing require lots of memory resources and processor time a variety of applications are still using ordinary computers to perform these tasks. Some examples are video surveillance applications, car license plate number identification, face recognition applications, etc. Another approach is to use an embedded system for this kind of applications.

Figure 7 shows a diagram of an embedded system. It consists mainly of an image sensor, some signal processing Integrated Circuit (IC) like an Application-Specific Integrated Circuit (ASIC), Digital Signal Processor (DSP), Field Programmable Gate Array (FPGA), Micro Controller Unit (MCU) or Reduced Instruction Set Computer (RISC) and optionally some external memory, other sensors, communications IC and interfaces.

The only part of the different approaches that can be changed is the signal processing unit.

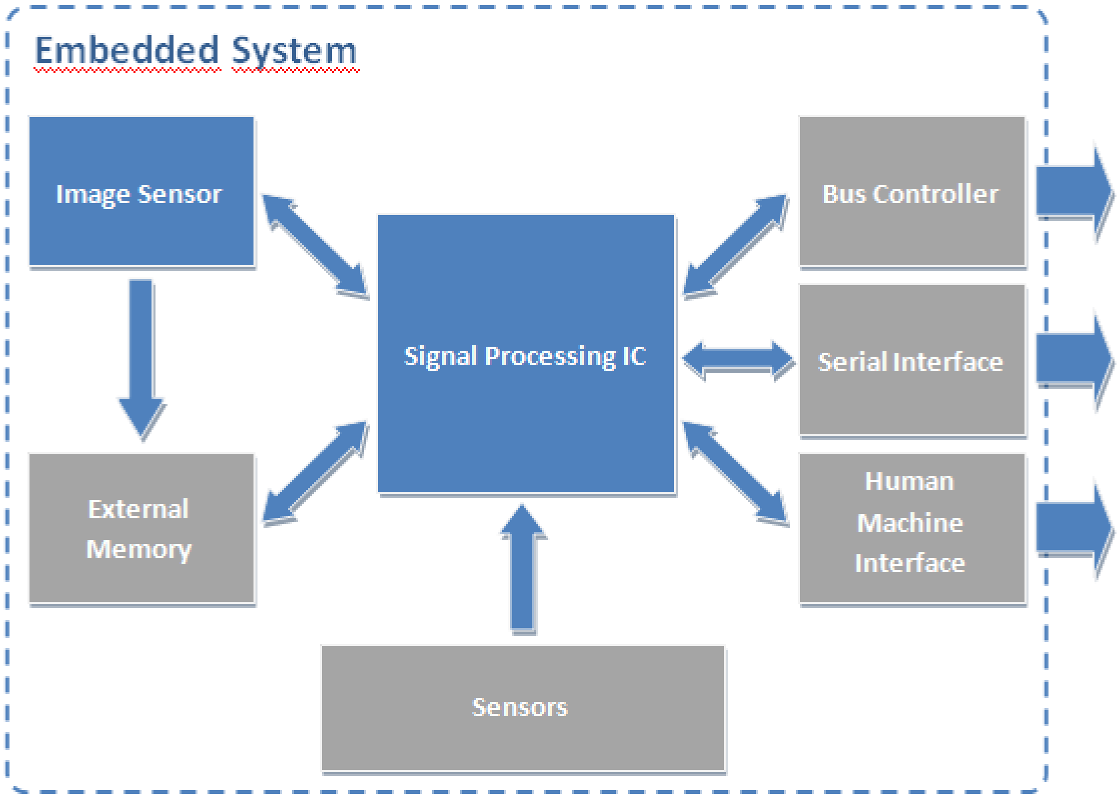


Figure 7. Embedded system for image processing.

The main advantage of this type of systems is that they are smaller than the computer approach so they can be installed in many other places. The drawbacks are the constraints in power consumption, memory, and processing speed. The emergence of more powerful microprocessors and microcontrollers has led to the gradual appearance of embedded systems for image processing.

In [12] an image sensor, a MCU and some laser diodes are used to measure object dimensions. They use an object with known dimension to calibrate the system and translate pixel information extracted from the laser lines in the image to real dimensions of the object. In [13]the authors have developed and embedded system based on a DSP to do fingerprint detection. The fingerprint database is stored in an external SDRAM memory. They also uses a keyboard and a display as the human machine interface. Another application of embedded systems can be found in [14]. This application consists on finding and recognizing car license plate numbers. An ARM processor is the core of this system that also contains an external memory, a keyboard, an LCD and different communications interfaces. In this application there are two different tasks. The first one consists on locating the car plate in the image. If this task is successful they extract the characters and perform a single-character processing to recognize the license number. Finally the recognized license number is shown in an LCD. Finally, in [15] an embedded system has been developed for face detection. In this approach authors use an FPGA as the core of the system. FPGA provides higher computational power but in the other hand they are much more expensive than MCU or DSP.

2.3. Previous work

Thinking Forward XXI has been working in a monitoring system for point machines since 2009. A collaboration between TMB and UPC allowed the development of such a system and the patent [3]. Students involved in the development founded a start-up in order to implant the system in TMB and in other potential customers.

The core element of the system is an acquisition device which is placed inside the point machine housing. This device captures a lot of parameters of the engine. Some of this parameters are the current, voltage, temperature, operation time, vibration, position of the lock bar and gap between the lock bar notch and the lock hammer.

The gap measure is estimated with a magnetic field produced by two magnets and two hall effect sensors. The magnets are fixed in lock bars, one in each bar. Hall effect sensors are fixed in the interior of the point machine housing. When the point machine is in normal position on sensor detects the magnetic field produced by the magnet fixed in the corresponding lock bar and produces an analog signal which is a function of the intensity of the magnetic field. This analog signal is translated with an Analog to Digital Converter (ADC) in a microcontroller. In this position, the other sensor is not detecting any magnetic field because the magnet which is fixed in the other lock bar is far away from the sensor.

This approach has several problems. The first one is that the analog signal from the sensor has to be calibrated in order to give a measure of the gap in millimeters. The calibration is done in the installation of the sensor. It consists on placing a collection of gauges of different known measures and annotate the correspondence between the ADC value and the gauge. In normal operation, when the microcontroller reads the ADC input it interpolates linearly the values to estimate the gap. For example, Table 1 shows a typical calibration for a sensor.

ADC Value	Gauge size (mm)
1120	0
1158	1
1174	2
1193	3
1205	4
1215	5

Table 1. Example of calibration points

If the ADC gives a measure of 1185 the system estimates the real gap to be 2.58mm as we can see in the following equation.

$$gap^* = 2 + (3-2) \times \frac{1185 - 1174}{1193 - 1174} = 2.58mm$$

The second problem is also related with the calibration. When operators perform the maintenance tasks on the point machines the calibration process has to be done again.

Usually, maintenance consists on changing some arrangements on the rails and this changes produce variations in the magnetic field measure. Sometimes maintenance tasks consists of changing the lock bars. In these cases the magnets have to be fixed again in new bars and, obviously, the calibration must be done another time.

Another problem is that the magnetic sensors are placed in the engine housing and connected with wires to the monitoring device. Sometimes the wires or the sensors themselves are broken during maintenance tasks.

Finally there is another issue that affects the actual measure of the gap. The magnetic field sensor measures the absolute value of the magnetic field in a particular axis. Usually, lock bars have more than one degree of freedom because of wear by friction. The movement caused by these additional degrees of freedom introduces errors in measures. Due to this fact it is possible that the lock bar does not move in the correct direction but in another one and this produces a variation in the magnetic field that hits the sensor.

In order to solve the above mentioned problems a new device has to be developed. In the following chapters a device based on an image processing technique will be designed and developed.

3. Fundamentals

In this chapter we will expose the theory of the methods we are going to use in the image processing algorithms. We can see a diagram of the proposed algorithm in Figure 32 in page 40. The first step of the algorithm consists on image binarization. Then, the binary image is filtered using a morphological filter. Finally the connected regions are found with a two scan algorithm. In addition to these three points camera calibration theory will also be explained.

3.1. Binarization

The first step in most image processing algorithms consists on binarizing the image in order extract objects from the background. Image binarization belongs to the group of the range transform operators. It is a non-reversible operation because it is based on two clippings.

Binarizing an image consists of clipping all the values below a given threshold to 0 and all values above the threshold to 1. It is also commonly known as thresholding. Figure 8 shows an example of a binarization mapping function S(r) where S is the value of the pixel after applying the function and r is the original intensity value of the pixel.

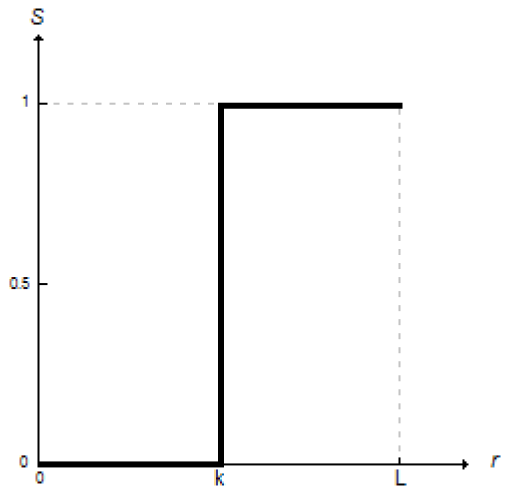


Figure 8. Binarization mapping function

In order to perform a good binarization it is very important to select an adequate threshold. We will use the Otsu method [17] to select the threshold. This method is based on the image histogram and it selects the optimum threshold by maximizing the inter-class variance of two classes. Classes are defined as C_0 which contains all pixels with a gray level between [0,1, … ,k] and C_1 which contains all pixels with gray level [k+1, … ,L] where L is the maximum gray level of the image and k is the threshold. In order to find the optimum threshold k^* it uses one discriminant criterion measure used in the discriminant analysis

$$\eta = \frac{\sigma_B^2}{\sigma_T^2}$$

where

$$\sigma_B^2 = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 = \omega_0\omega_1(\mu_1 - \mu_0)^2$$

and

$$\sigma_T^2 = \sum_{i=1}^{L} (i - \mu_T)^2 p_i$$

p_i is the probability of a pixel to have gray level i and ω_i and μ_i are the zeroth-order and first-order cumulative moments of classes defined with kth level. σ_B^2 and σ_T^2 are the between class variance and the total variance of levels respectively. Maximizing the discriminant criterion η is equivalent to maximizing σ_B^2 because σ_T^2 does not depend on threshold k. Therefore the optimal threshold k^* is found using a sequential search of the maximum of σ_B^2 for all possible values of k=[0, … ,L)

3.2. Mathematical Morphology

Mathematical Morphology is used in the field of image processing to analyze images from the geometrical point of view. It is based on set and lattice theory.

In order to define lattice theory we have to review the superposition principle. Once we assume a certain superposition principle, there is a natural operator. The superposition principle is intrinsic of the signal creation. For classical linear functions the superposition principle is the linear superposition principle in the vector space. The natural operator in this linear space are the vector addition and the scalar product. Figure 9 illustrates the linear superposition principle.

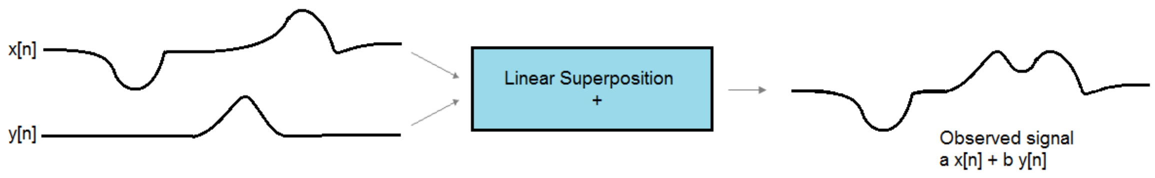


Figure 9. Linear superposition principle

In the linear space the basic sequence is the unit impulse. This sequence determines the natural operator which is characterized by the impulse response and can be computed with the convolution.

Linear superposition is not good for images as we can see in Figure 10. Applying linear superposition means that we have to linearly combine different objects in the image. This is as if the objects were partially transparent. This is not true due to objects near the camera occludes objects in the background.

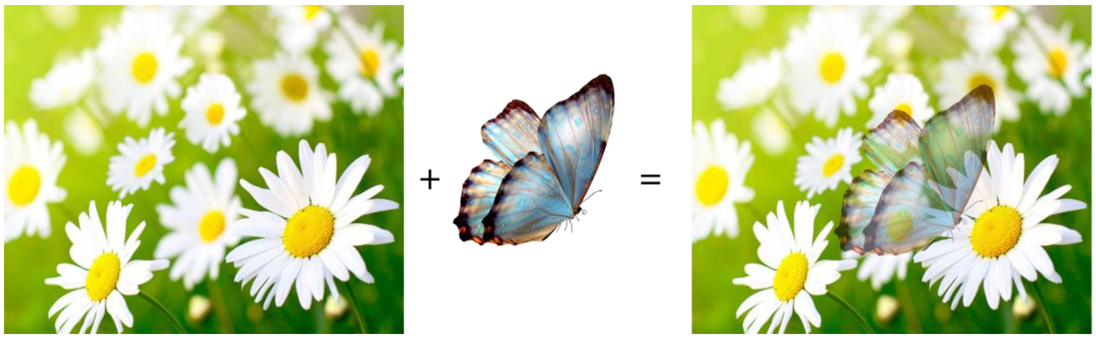


Figure 10. Linear superposition in images

Lattice is another mathematical structure, like the vector space in linear superposition, which is characterized because it has partial order relation, ≤, and two dual operators, supremum, ∨, and infimum, ∧.

Partial order relation, ≤, means the following:

$\forall\, x, y, z \in \mathbb{R}$

- $x \leq x$. The partial order relation is a reflexive relation. This means that the relation holds true for every element x in the set.
- $x \leq y, y \leq x \Rightarrow x = y$. It is asymmetric. There is no pair of different elements of the set each of which is related by ≤ to the other. In other words, the only way for that the partial order relation holds true for x, y and y, x is that x=y.
- $x \leq y, y \leq z \Rightarrow x \leq z$. It is transitive. This means that if x is related to y and y is in turn related to z, then x is also related to z.

The infimum of a subset S of a partially ordered set T is the greatest element of T that is less than or equal to all elements of S. The infimum is also known as the greatest lower bound. Formally, the infimum of a subset S of a partially ordered set P is an element a of P such that:

- $a \leq x\ \forall x \in S$. a is a lower bound, and
- $\forall y \in P, if\ \ \forall x \in S, y \leq x \ \Rightarrow y \leq a$. a is larger than any other lower bound.

The supremum is the dual concept of infimum. The supremum of a subset S of a partially ordered set T is the lowest element of T that is more than or equal to all elements of S. It is also known as the least upper bound. Formally, the supremum of a subset S of a partially ordered set P is an element a of P such that:

- $x \leq a, \forall x \in S$. a is a upper bound, and
- $\forall y \in P, if\ \forall x \in S, x \leq y \ \Rightarrow a \leq y$. a is smaller than any other upper bound.

In order to use mathematical morphology for image processing we have to introduce the concept of the lattice of functions. A lattice of functions is a set of functions that has a partial order relation. This means that x ≤ y holds if and only if $x[n] \leq y[n]\ \forall n$. Figure 11 illustrates the concept of lattice of functions. At the left we can see two functions that fulfills the partial order relation while the functions f and g in the image at the right does not fulfill the relation.

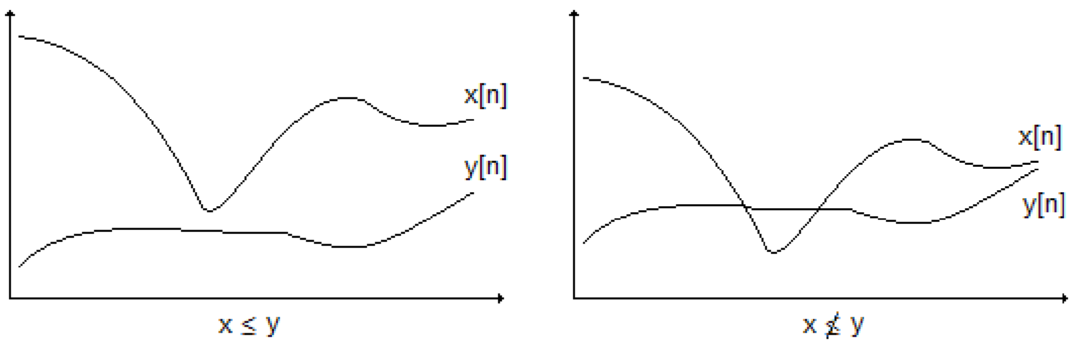


Figure 11. Lattice of functions

The operators supremum and infimum can also be applied in the lattice of functions.

- Supremum: $z = x \vee y$, this is $z[n] = \max(x[n], y[n])\ \forall n$
- Infimum: $z = x \wedge y$, this is $z(n) = \min(x[n], y[n])\ \forall n$

Figure 12 shows the concept of the supremum and infimum in the function lattice. On the left we have the supremum of the functions x[n] and y[n] marked in red. On the right we have marked in red the infimum of the functions x[n] and y[n].

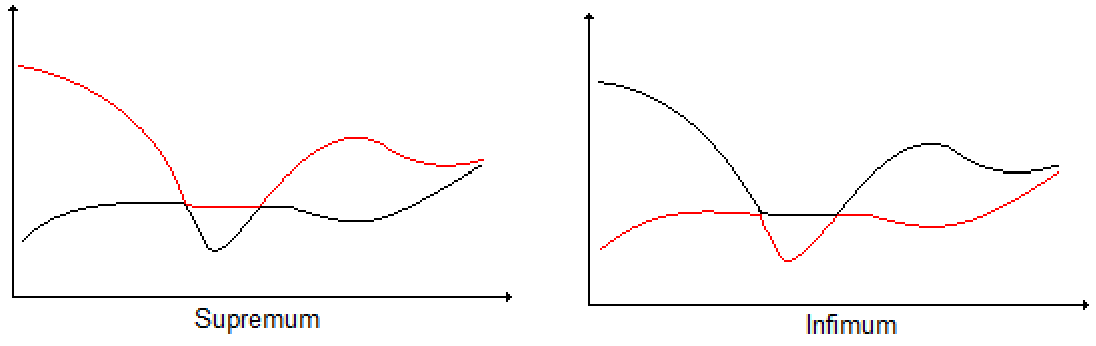


Figure 12. Supremum and infimum in the function lattice

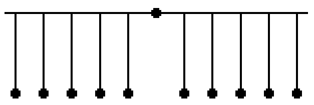
On important property of the supremum and infimum is that they are dual.

$$x \wedge y = -[(-x) \vee (-y)]$$

Because of this, all morphological operators will appear in pair.

Similarly to the unit impulse in the linear space, the point is the basic sequence in the lattice structure.

$$\delta_L[n] = \begin{cases} 0, if\ n = 0 \\ -\infty\ otherwise \end{cases}$$



Any function in the lattice space can be decomposed using the point. In order to recover the original function we have to take the supremum at each point.

$$x[n] = \bigvee_{k=-\infty}^{\infty} (x[k] + \delta_L[n-k])$$

Now we can design the natural operator.

- It must preserve the lattice structure: $x \leq y \Rightarrow \psi(x) \leq \psi(y)$
- It must be compatible with the superposition principle, and
- It must be translation invariant

$$y[n] = \psi\{x[n]\} = \psi\left\{\bigvee_{k=-\infty}^{\infty} x[k] + \delta_L[n-k]\right\}$$

If we take into account the superposition principle, we can commute the operator with $\vee$ and +

$$y[n] = \bigvee_{k=-\infty}^{\infty} (x[k] + \psi\{\delta_L[n-k]\}$$

Applying the translation invariant property we have that $b[n] = \psi\{\delta_L[n]\}$, and then

$$y[n] = \bigvee_{k=-\infty}^{\infty} (x[k] + b[n-k])$$

This is the dilation operator and it is similar to a nonlinear convolution. It is characterized by b[n] which is called the "structuring element". The dilation is denoted as

$$\delta_b\{x[n]\} = x[n] \oplus b[n]$$

The erosion operator is the dual of the dilation. Erosion is similar to a nonlinear correlation and it is also characterized by the structuring element b[n]. Formally, the erosion is

$$\varepsilon_b\{x[n]\} = x[n] \ominus b[n] = \bigwedge_{k=-\infty}^{\infty} (x[k] - b[k-n])$$

In order to compute erosion and dilation in image processing we use a flat structuring element. This means that the possible values of $b[n] \in \{0, -\infty\}$. Flat structuring element allows us to perform simple computations, only the min or max of the signal. Furthermore the output is one of the input samples, this means that it does not modify the dynamic range of the output. Another advantage is that it preserve the contrast around edges.

The locations where b[n]=0 defines a window and the dilation consists in computing the maximum of gray level values below the window. Dually, the erosion consists in computing the minimum of gray level values below the window.

The properties of the dilation and erosion are the following:

- Increasing. They preserve the lattice structure

  $x \leq y \Rightarrow x \oplus b \leq y \oplus b$

  $x \leq y \Rightarrow x \ominus b \leq y \ominus b$

- It is distributive. We can compute the dilation of the supremum of two signals by computing the supremum of the signals after applying the dilation:

  $(x \vee y) \oplus b = (x \oplus b) \vee (y \oplus b)$

  $(x \wedge y) \ominus b = (x \ominus b) \wedge (y \ominus b)$

- Composition. Iteration of erosion and dilation is equivalent to use a larger structuring element.

  $x \oplus b1 \oplus b2 = x \oplus b, \;\; b = b1 \oplus b2$

  $x \ominus b1 \ominus b2 = x \ominus b, \;\; b = b1 \oplus b2$ (note that the resulting b is compute with a dilation)

- Extensive or anti-extensive. An operator is extensive if the output of the operator is greater than or equal to the input signal for all x, this is $x \leq \psi(x) \;\; \forall x$. It is anti-extensive if the output of the operator is less than or equal to the input signal for all x, this is $\psi(x) \leq x \;\; \forall x$.

  If the space origin belongs to the structuring element, b[0]=0, then the dilation is extensive and the erosion anti-extensive.

Figure 13 shows an example of applying the dilation and erosion operators to a binary image. For this example, the structuring element is a diamond of size 7x7 pixels with the origin in the center of the structuring element. As we can see in this example the dilation thickens the objects on the foreground and it removes dark components (min) inside the foreground objects. In the other hand, erosion flattens the objects in the foreground and removes bright components (max) in the background. We can observe that both operators preserve the form of the edges of the image. This represents a great difference between morphological operators and frequency domain operators.

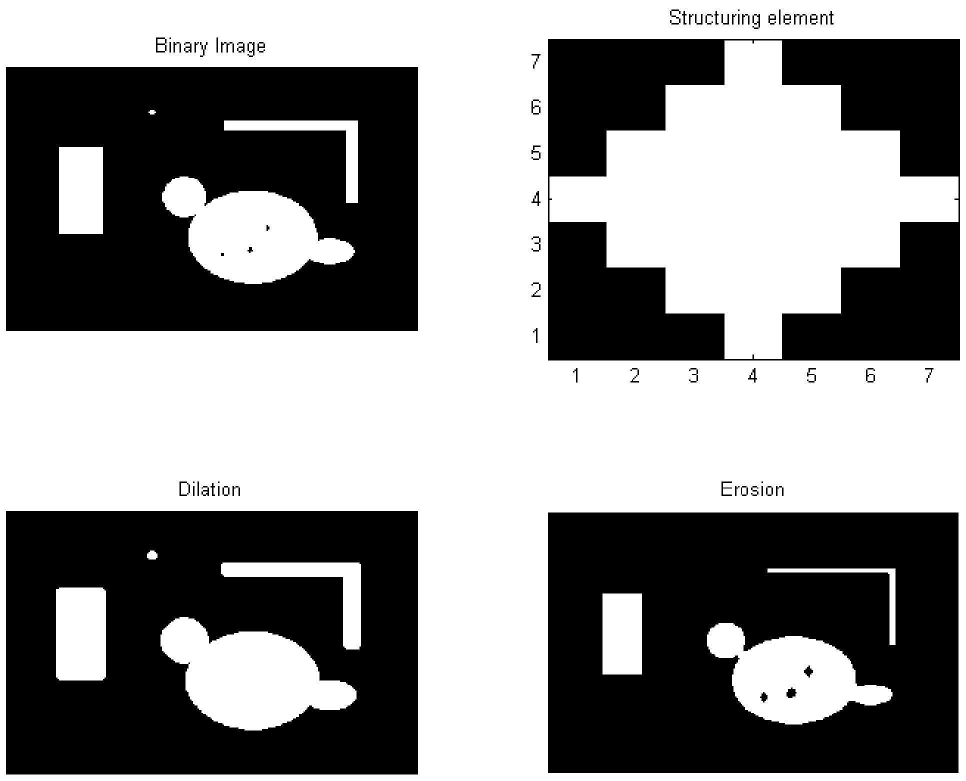


Figure 13. Erosion and dilation with a diamond structuring element

The combination of dilation and erosion allows us to build new morphological operators. As we have stated previously, the erosion operation is useful for removing small objects in the background. However, it has the disadvantage that all the remaining objects shrink in size as we have seen in Figure 13. This effect can be avoided by applying a dilation after the erosion with the same structuring element. This combination of operations is called an opening:

$$\gamma_b(x) = (x \ominus b) \oplus b$$

The dual operator of the opening is the closing operator. It consist of concatenating a dilation with an erosion with the same structuring element:

$$\varphi_b(x) = (x \ominus b) \oplus b$$

The main application of opening consists on removing small objects in the image while preserving the contours of the objects that remain in the image. The main application of closing is background simplification. Closing fills small gaps in the objects and gather together close objects in the image. Figure 14 illustrates these concepts.

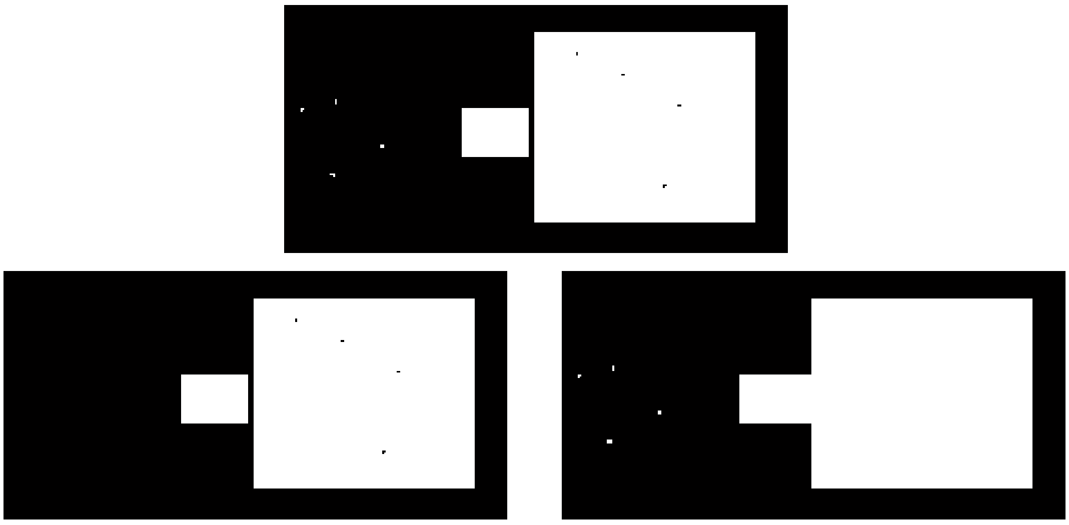


Figure 14. Opening (left) and closing (right) of a binary image

The main properties of the opening and closing operators are:

- They are increasing, this is

$$if\ x \leq y \quad \Rightarrow \gamma_b\,(x) \leq \gamma_b(y)$$
$$\Rightarrow \varphi_b(x) \leq \varphi_b(y)$$

- They are idempotent. If we apply more than once the operator is the same as applying it one time. Formally,

$$\psi\big(\psi(x)\big) = \psi(x)$$

This two properties allows us to define the concept of morphological filter. As they are increasing they preserve the lattice structure and furthermore the filtering effect is controlled because we have to filter just once. Combinations of openings and closing are also morphological filters. Figure 15 shows the previous image filtered with an opening after a closing. The effect is that we have deleted small objects in the image, we have filled the holes of the object and we have simplified the background by gathering together two close objects.

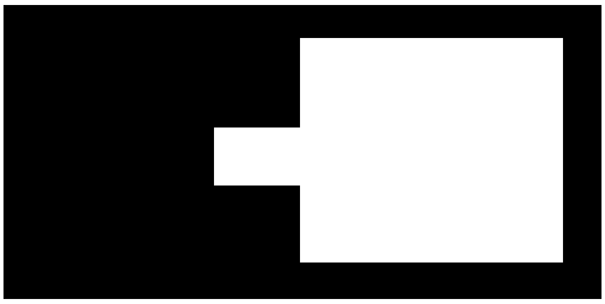


Figure 15. Morphological filter

3.3. <u>Connected components</u>

Although binarization and morphological filtering takes into account neighbor pixel values, they actually apply transformations on individual pixels. Usually, an image contains objects and so we need some technique to identify the pixels corresponding to these objects. Connected components theory tries to identify and separate pixels according to the real world object they belong.

For instance, in Figure 16 we can distinguish 10 objects corresponding to 6 letters and 4 geometrical shapes. This is a binary image, that is, all pixels corresponding to an object (foreground) have the value 1 and the other pixels corresponding to the background have value 0.

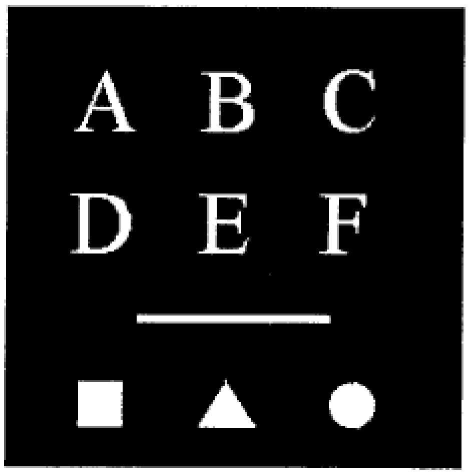


Figure 16. Image containing 10 objects

A pixel *p* at coordinates (x, y) 4-neighbors denoted as $N_4(p)$. This set is formed by the pixels at locations $(x+1, y), (x, y+1), (x-1, y)$ and $(x, y-1)$. We can define another set of 4 pixels denoted as $N_D(p)$ which contains the diagonal neighbors of p whose coordinates are $(x-1, y-1), (x+1, y-1), (x-1, y+1)$ and $(x+1, y+1)$. The union of sets $N_4(p) \cup N_D(p)$ is the set $N_8(p)$ which represent the 8-neighbour of pixel *p*. Two pixels *p* and *q* are said to be 4-adjacent if $q \in N_4(p)$. Similarly *p* and *q* are 8-adjacent if $q \in N_8(p)$. Figure 17 represents neighbors of pixel *p.*

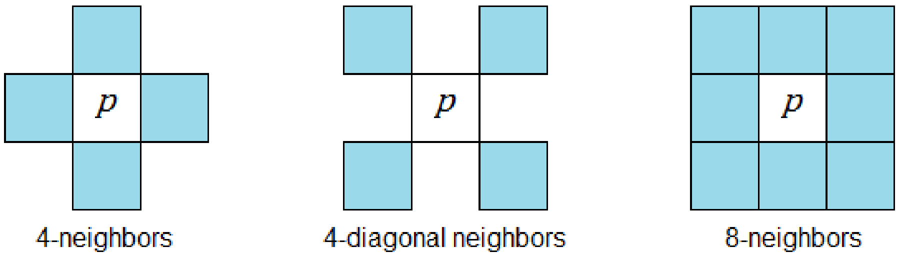


Figure 17. Neighbors of pixel *p*

We can define a *path* between p_1 and p_n as a sequence of pixels $p_1, p_2, \dots, p_{n-1}, p_n$ such that p_k is adjacent to p_{k+1}, for $1 \leq k < n$. A path can be 4-connected of 8-connected depending on the definition of adjacency used.

Two foreground pixels *p* and *q* are said to be 4-connected if there exists a 4-connected path between them, consisting completely of foreground pixels. They are 8-connected if there exists an 8-connected path between them. Figure 18 illustrates these concepts. Gray pixels represents the background while white pixels are part of the foreground. On the left there exists a 4-connected and 8-connected path between *p* and *q.* In the central figure there only is a 8-connected path because the 4-connected condition is not fulfilled in the central pixel. On the right image there exists a 4-connected path and two 8-connected paths.

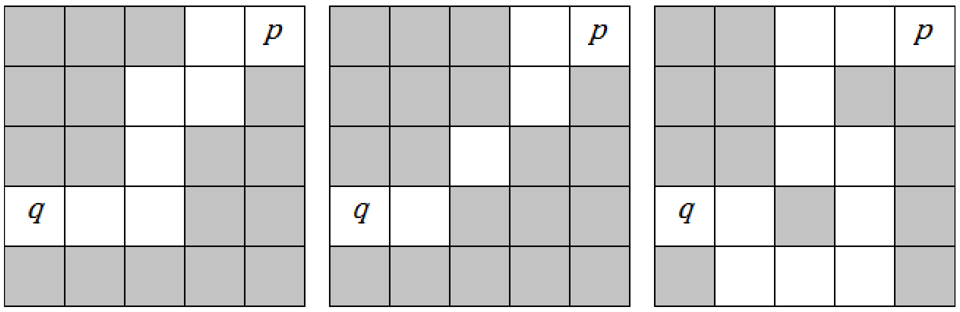


Figure 18. Different types of path between two pixels.

A connected component C_c is the set of all foreground pixels that are connected to each other, this is, there exists a path between any pair of $p, q \in C_c$. Note that connected component is defined in terms of a path, and the definition of the path depends on adjacency, therefore we need to define the type of adjacency to find the connected components of an image.

3.4. Camera calibration

4. Methodology / project development

In this chapter we will explain the key aspects of the design. We will start by establishing the system specifications that fulfills the goals presented in Section 1.3. Then we will continue explaining the hardware design which involves not also the electronics but the optical element and the housing of the device. After that a communications challenge will be presented and we will explain a custom communications protocol to solve it. Finally the image processing algorithms will be discussed.

4.1. System specifications

The device must be designed taking into account the goals stated in Section 1.3 and the size constraints.

- First of all the device must be easy to install. There are only 3 daily hours to perform the maintenance tasks and maintenance teams are small according to the size of the infrastructure to maintain. The most easy to install the device the better. This should be taken into account in the hardware, especially in the box design, and in the firmware design.
- The device needs to be robust in two senses. The first one relates directly with the hardware design because it has to be installed inside a point machine which is subject to vibration, temperature changes, dust and humidity. The box of the device must resists this unfavorable conditions. The other sense refers to the repeatability and confidence of the measurements and it is related with the software design and implementation. The error rate of the device should be less than 5%, including bad measures and unknown measures when the lock bar is in its correct position.
- The minimum total gap to measure is of 5 millimeters. Maintenance operators want to know where the gap i less than 10% in order to plan a task to correct the position of the bars. The precision of the measurement should be less than 0.5mm and it is desirable to be 0.25mm. This should be taken into account to determine the resolution of the CCD and image processing algorithms.
- Another constraint of the system is the execution time that consists on the image capture plus the image processing times. The image capture time is determined by the hardware (the image sensor, the microcontroller and the memory latency). The image algorithm determines the processing time. As it has been explained in Section 1.3 the device has to two modes of operation. In normal operation it gives a measure every 10 seconds. In general, the execution time could be up to 10 seconds. In continuous operation the device must provide at least one measure per second. This has to be considered in the design of the image processing algorithm.
- Power consumption has also be taken into account. The wire length between the control cabin and the point machine location can sometimes be quite long, up to 1Km. The device is powered by a DC voltage up to 24V with a cable of 1.5mm^2. The resistivity of the copper is

$$\rho = R\frac{S}{l} = 1.71 \times 10^{-8}\Omega m$$

as we can see in [16]. S is the section of the conductor in squared meters and l is the wire length in meters. We can compute the total wire resistance as

$$R = \rho \frac{l}{S} = 11.86\Omega$$

If we take, for instance, that the device consumes 150mA, the total drop of potential at the device is

$$E = I \times 2R = 3.56V$$

Typically the same cable is used to power more than one device so the power consumption represents a constraint on the design. This constraint relates not only with the hardware but also with the firmware of the device. If the execution time is so long that the device is always running then the power consumption increases and it probably does not fulfill the power consumption constraint.

– Size constraints. The device has to fit in the interior of the engine housing. There is a protective element which can be used to hang it. Its position is the best for the device because it is just above the region of interest and the vertical distance from the bars to the position of this element is suitable. If we design the device to be hung from this element there is a size constraint. Figure 19 shows this element. First of all, we want to device to be as small as possible because it should not interfere with the visual inspection of the gap. Furthermore this device could be used in another kind of engine and the smaller the better to find a suitable position for it. Secondly the space between the protective element and the cover of the engine is about 10 cm. The cover is not fixed to the engine housing so it has some freedom when an operator opens or closes it. If the device is to close to the cover the likelihood of breaking it increases. So we must design it to be at least 2cm away from the cover.

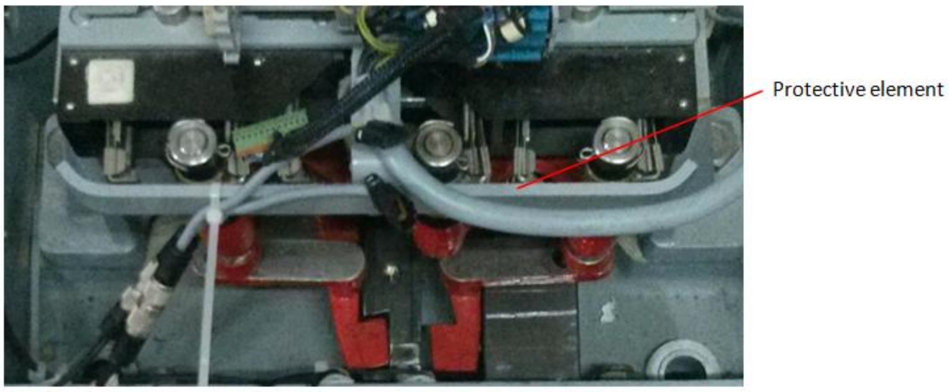


Figure 19. Protective element where the device can be hung.

4.2. <u>Hardware Design</u>

4.2.1. System architecture

The most important element of the product is the image sensor and a suitable lens. The key point in the hardware design is that we have to develop a product within the minimum possible time and with the available resources in Thinking Forward XXI. In order to simplify the development, the first option to evaluate is to buy a module which includes the optical sensor and the lens. We have rejected this approach because we have found that manufacturers like SHARP or TOSHIBA are not interested in selling little quantities of their integrated modules and furthermore almost all available modules in the market have a CCTV output that it is not useful for our application.

The second option we have evaluated is to use an image sensor with a microcontroller. There are some reasons why we are going to develop this device using this option. First of all, the team in Thinking Forward XXI which is going to design and develop the device has some expertise in developing hardware based on microcontrollers. Secondly, microcontrollers are cheaper than FPGA or ASICs. And finally we have a first prototype designed with this approach. In addition, the microcontroller approach has been used by Pauli et al in [12] so we know in advance that it is a feasible solution. Figure 20 is a diagram of the system architecture. In the following subsections we are going to explain each of the modules.

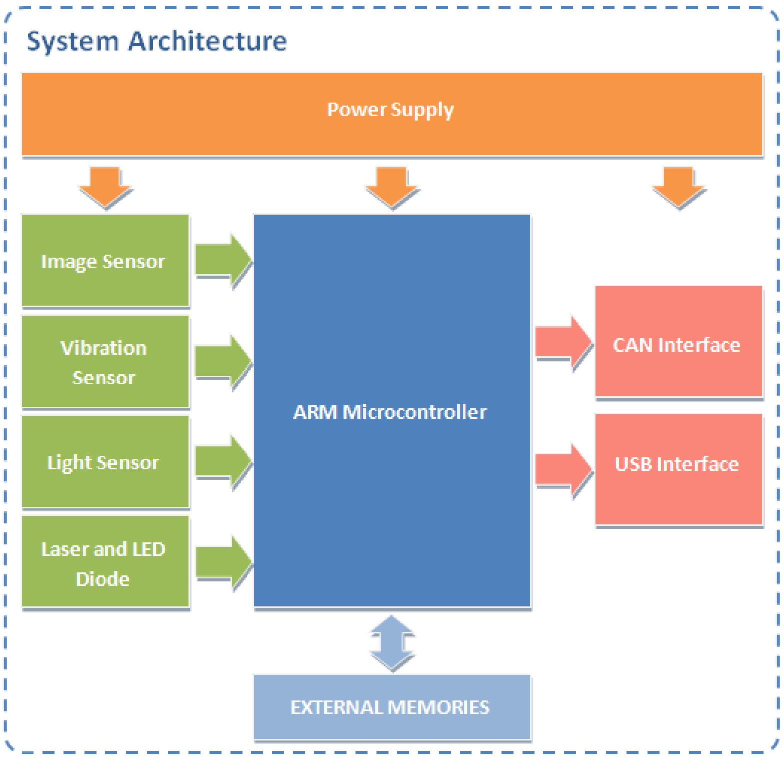


Figure 20. System architecture

4.2.1.1. Microcontroller

The microcontroller must have a CAN interface and a Camera interface. The CAN interface is compulsory because the device has to be installed inside the engines. Usually there are free wires from the control point to the engine which are the only way to establish the communication. The new device has to be integrated in a system which uses these wires to make a CAN bus and send information over it. The best choice is to use the same bus to send the engine position, gap measurements and images. The camera interface is also compulsory to connect the microcontroller with the image sensor. The microcontroller we have selected is an ARM Cortex-M4 from ST. More specifically it is the STM32F407 which goes up to 168MHz. We can see the electrical diagram of the component in Figure 24.

4.2.1.2. Image Sensor

In order to select the image sensor we have to take into account the resolution we want to achieve. The area we have to analyze is more or less a square of w=12cm side. The resolution we want to achieve is about 0.25mm as we have said in Chapter 0. We can compute the minimum resolution of the sensor in pixels to be

$$min\, res = \frac{120}{0.25} = 480px$$

Nor the maximum resolution neither the number of channels are important parameters for this application. Although the maximum resolution should be taken into account because it is related to the processing time, it is not a restrictive parameter in this application because usually image sensors can be configured to the desired resolution. Despite the application only requires one channel images (gray level images), color sensors have the same price as gray sensors.

Another important aspect to take into account is the availability of the sensor. After talking with some sensor manufacturers and distributors we have found that the best choice is Aptina. Although you have to sign a non disclosure agreement (NDA), the sensors can be bought in any component provider like RS or Farnell. After signing the NDA Aptina provides useful datasheets and developer guides that allows you to configure the sensor.

Taking this into account we have requested information about an Aptina and an Omnivision sensor and finally we have selected the MT9M131C12STC sensor from Aptina because the documentation is much better. It is a 1.3Mpx color sensor and it can be configured through an I2C interface which is also available in the selected microcontroller. In Figure 25 we can see the electrical diagram of the device and the signals connected to it.

4.2.1.3. Light Pattern

In Section 4.3.2.5 and 4.3.2.6 we will explain how we estimate the position and compute the gap. The solution which performs better and allows us to reduce the processing time is to use some kind of structured light.

We have considered using a single laser with a Diffractive Optical Element (DOE). DOE's allows to control the beam's shape of the laser light allowing us to have different patterns with the same laser component.

We have tested 3 different DOE elements which patterns can be found in Figure 21. In this figure we can also see the measures of the patterns produced by the elements. In Table 2 we can find the pattern angles for each element.

The DOE have to be placed, approximately, at 150mm of the plane we have to measure. The computed size of the projected pattern at this distance can be also found in Table 2. To compute the pattern size we have applied the following trigonometric relation considering that the laser beam goes from a single dot to the interest plane following a straight line.

$$a = 2 \times \tan\left(\frac{\alpha}{2}\right) \times h$$

The same computation has been applied to the other measures b,c and d. The region of interest is about d=60mm and b=80mm. Taking into account this measures the DOE which fits better for our application is the 11 lines one.

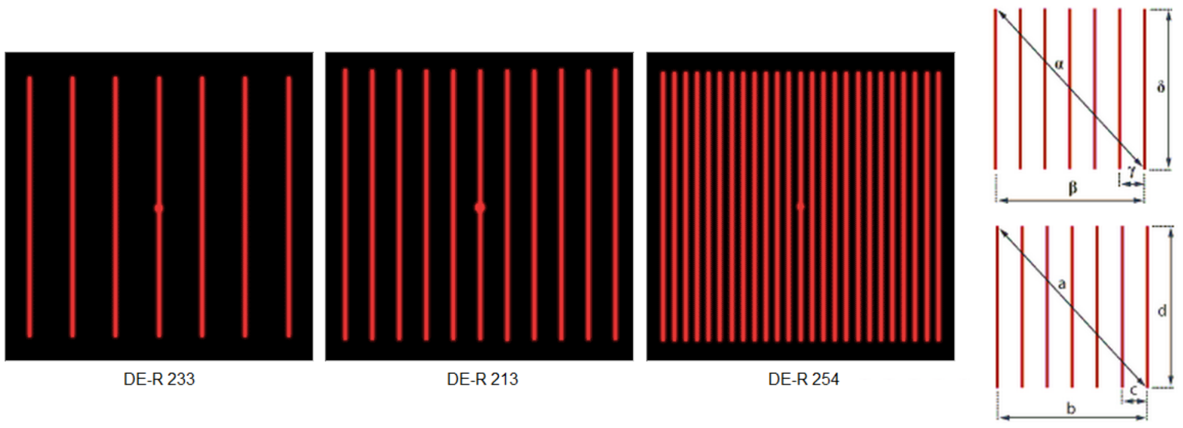


Figure 21. DOE Patterns and pattern measures

DOE Item	Description	α	β	γ	δ	a (mm)	b (mm)	c (mm)	d (mm)
DE-R233	7 lines	30º	22º	3.6º	22º	80.4	58.3	9.4	58.3
DE-R213	11 lines	41.8º	30.3º	3º	30.3º	114.6	81.2	7.9	81.2
DE-R254	25 lines	36º	26º	1.09º	26º	97.5	69.3	2.9	69.3

Table 2. DOE Pattern angles and computed pattern size at 150mm distance

This solution has a main drawback which is the cost of the DOE. A single DOE element costs up to 40€ while a laser with a single line diffractive element costs less than 2€. As we have state in the goals, section 1.3, one requirement of the system is that it has to be inexpensive. So we have decided that the better solution for this device is to mount two laser diodes with a single line pattern each one.

This solution has another advantage. The lasers can be placed symmetrically one at each side of the image sensor. Then, when an operator has to install the device he or she can center it only by looking at the projected lines. The device will be correctly installed if the two lines are centered in the lock bars.

4.2.1.4. Schematic

Some aspects have to be taken into account to design the schematic.

- The device will be supplied with the same cable than the other monitoring device and therefore the input voltage to the circuit can be from 8V up to 24V. The input power to digital components must be conditioned according to their datasheets. Usually this means that decoupling capacitors must be placed between power supply and ground. Figure 27 shows the power supply circuit and the conditioning circuit. The LM2937IMP-5.0 is the first linear regulator. The input voltage to this component must be between 6V and 26V. If the input voltage is inside this range the output voltage is stable at 5V. The second LDO, TPS71828-30 is a dual regulator that gives two outputs. One at 3.0V and the other at 2.8V. The Aptina image sensor have to be powered with an input voltage of 2.8V. All the other components are powered with the 3.0V output.
- The communication has to be done through a CAN bus so we must use a CAN transceiver. Figure 22 shows the CAN transceiver circuit.
- The image sensor has a resolution of 1290x1024 pixels. The pixels are coded in YCbCr so each pixels needs 2 bytes to be stored. We need at least 2.5MB to

store the image. Microcontrollers have much less memory than this so we will need an external RAM memory to store one or more images. In Figure 26 we can see this external memory.

- A photodiode will be included in the design in order to detect external illumination. One of our customers wants to have information about when and how long the point machines are opened for maintenance. This sensor will able us to detect this particular situation. Another vibration sensor will be also placed in the device. An important event for our customers is to monitor the reaction of the lock bars when a train passes over the point machine. We will use the vibration information in order to detect trains circulation. When a train is detected the device enters in a special mode and analyses the images continuously. Additionally an external USB memory can be plugged in the device to store short videos.

The schematic has been designed taking into account this points and the previous ones stated in points **¡Error! No se encuentra el origen de la referencia.**, 4.2.1.1 and 4.2.1.2. **¡Error! No se encuentra el origen de la referencia.**Figure 22 to Figure 27 show the detail of the design.

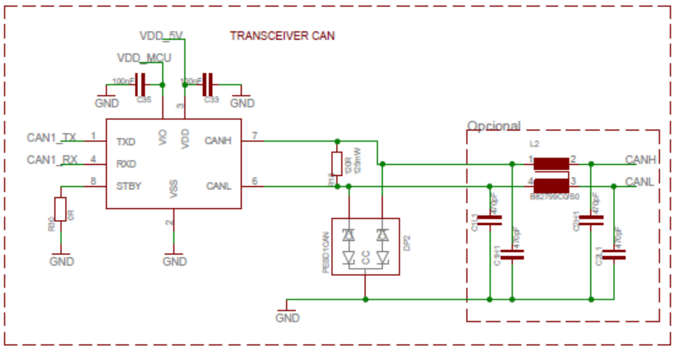


Figure 22. Schematics. CAN Transceiver

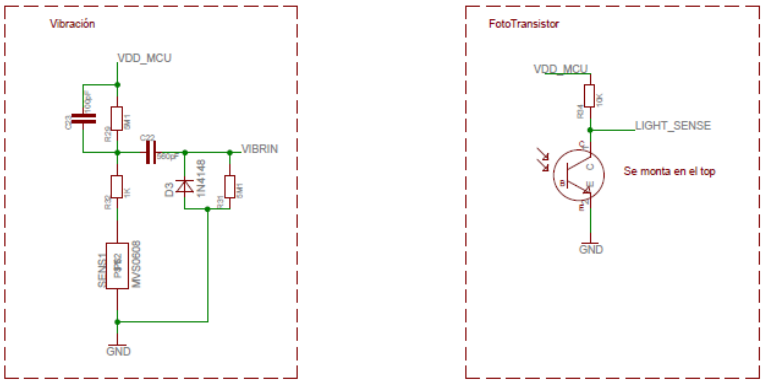


Figure 23. Schematics. External Sensors, vibration and light

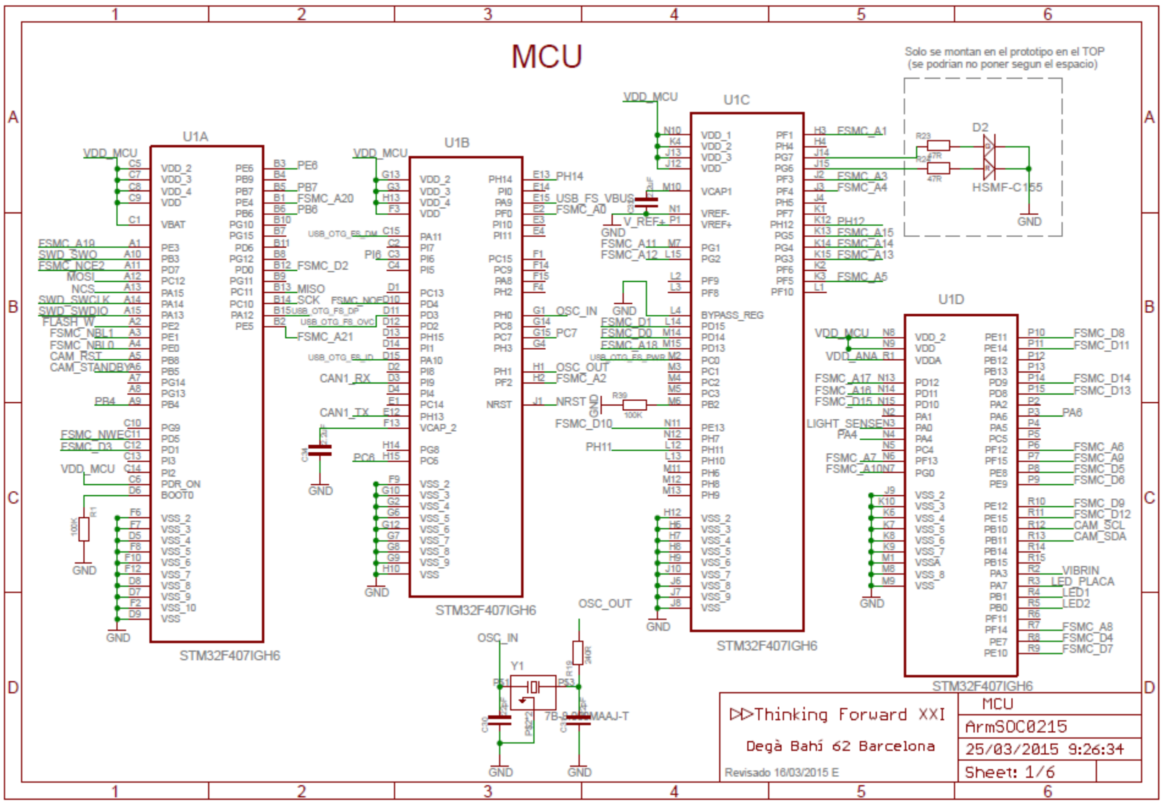


Figure 24. Schematics. Microcontroller

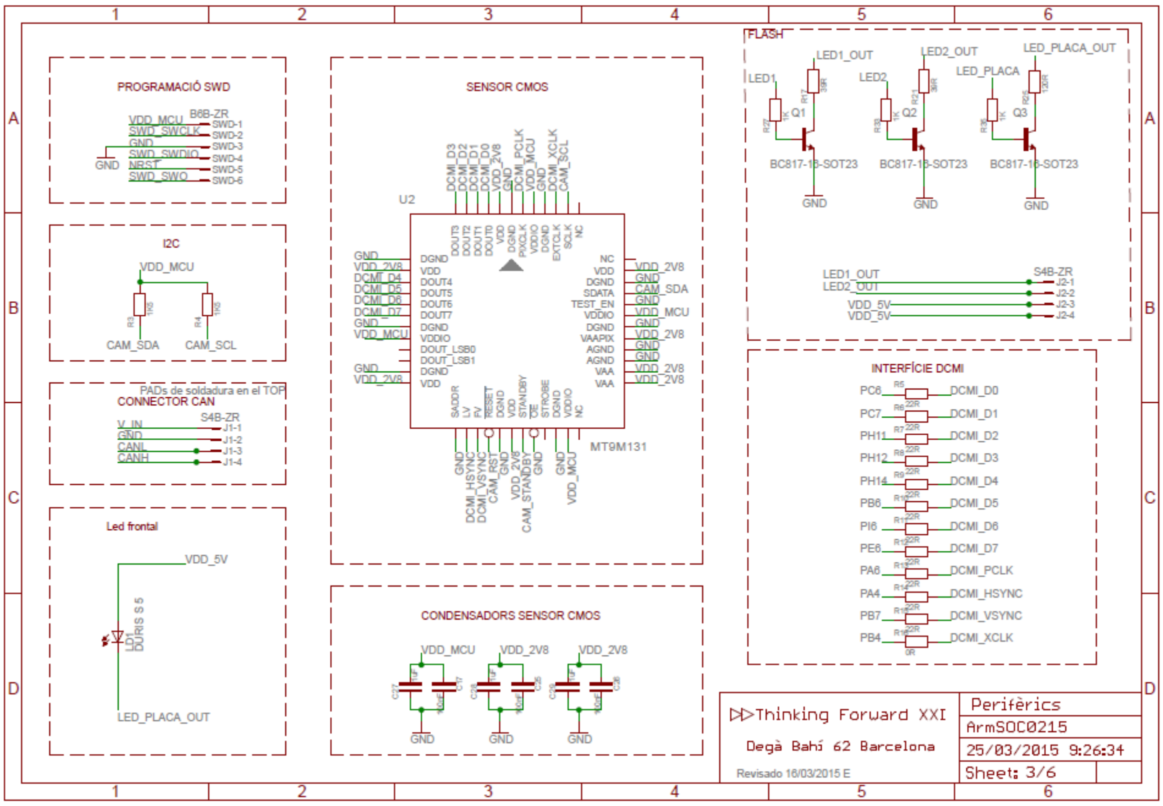


Figure 25. Schematics. Image Sensor

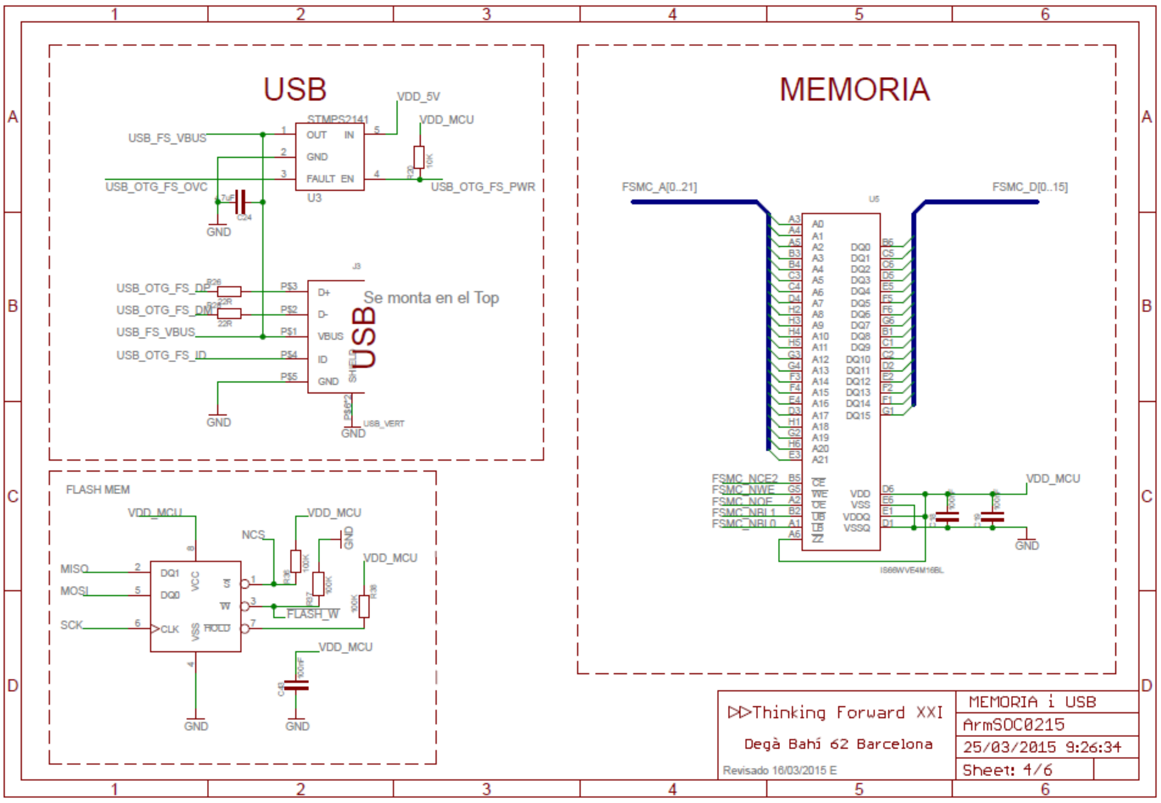


Figure 26. Schematics. External Memories and USB interface

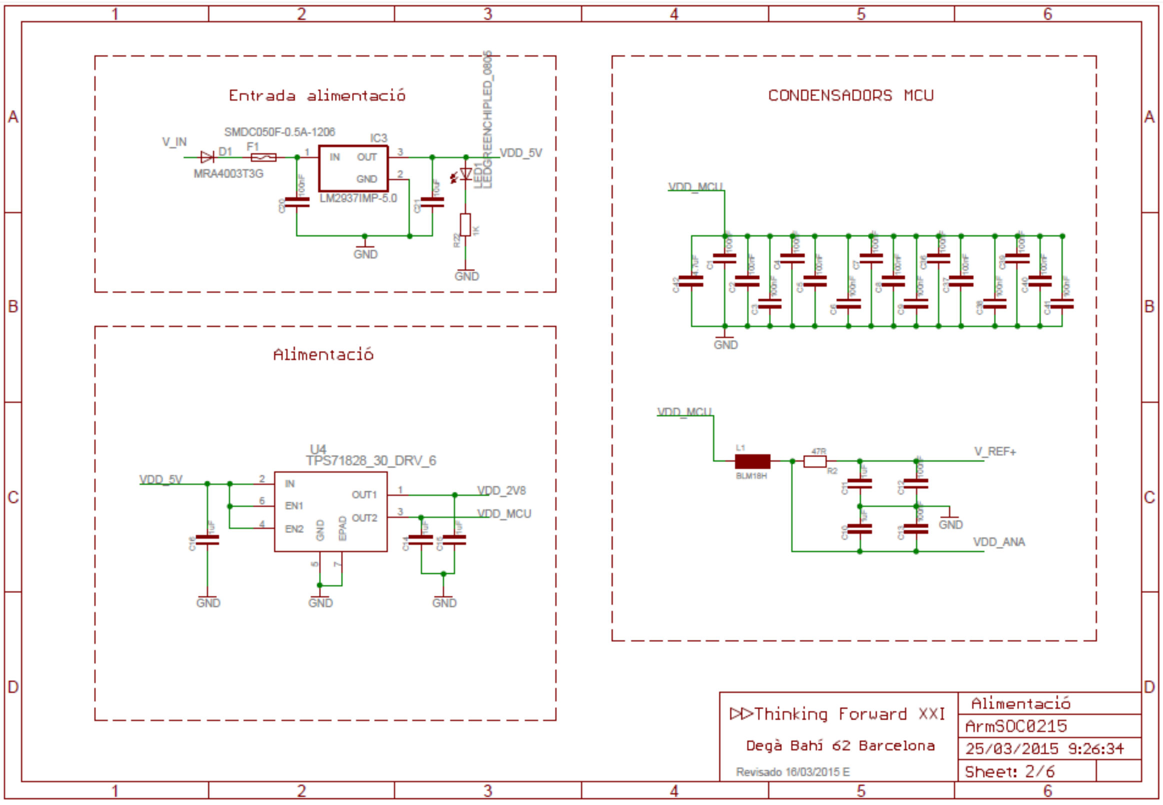


Figure 27. Schematics. Power Supply circuit

4.2.1.5. Layout Design

The design of the layout has to be done taking into account, basically, the size constraints and the electromagnetic compatibility of the device. Due to there is not any expert in layout design in the project this have been outsourced.

For completeness of the documentation Figure 28 shows the final layout design and Figure 29 is an image of the final PCB with all the components. In the first design we had to make some adjustments.

- The position of the Flash has been changed. In the first design the LED was placed at the bottom of one of the laser diodes.
- The position of the photodiode has also been changed. At the beginning it was placed in the side of the board that points to the interior of the point machine. We have changed this component because it is better to place it in top the side of the board because in this side it receives more light when the housing top is opened.
- We have introduced a non-volatile Flash Memory in order to store some configuration parameters. At the beginning we considered the possibility of using the internal Flash of the microcontroller. We have decide to move it to an external component because of two reasons. The first one is that the writing to the internal flash is very slow. The second reason is because the external component is more reliable. Using a separate component we can also store the microcontroller firmware to reprogram it when necessary.

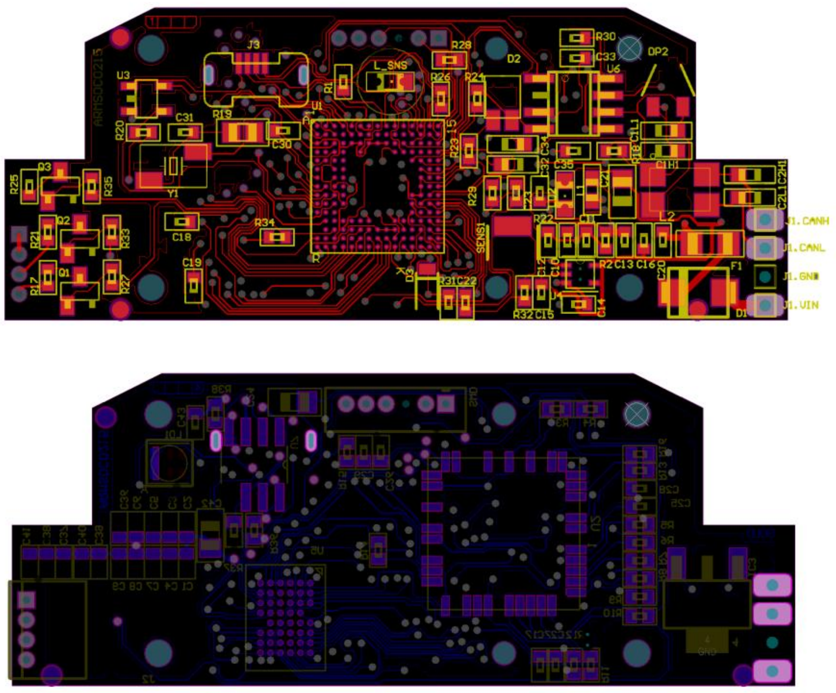


Figure 28. Final Layout Design, top and bottom

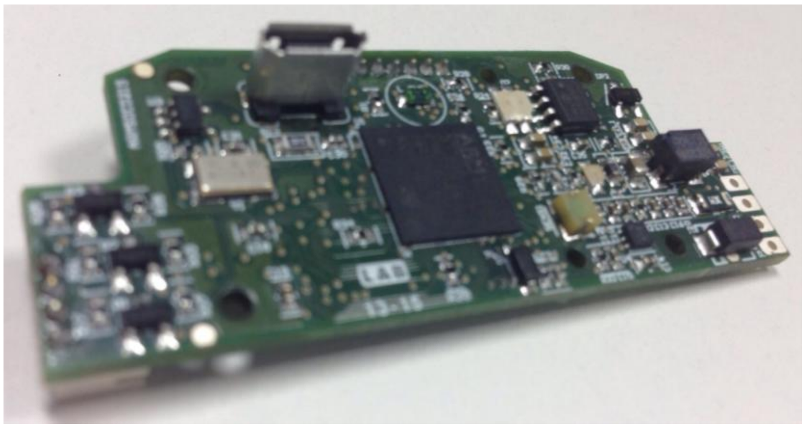
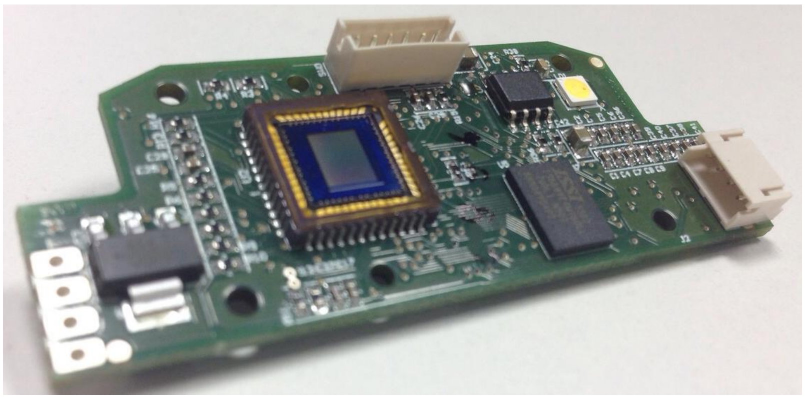



Figure 29. Final Printed Circuit Board, top and bottom

4.2.2. Optical element

We are now focusing in the optical element. To select the lens we have to focus on the geometry of the problem. The device is going to be placed approximately D=15cm over the image plane. As we have said previously, the area we have to analyze is more or less a square of w=12cm side. We can compute the angle of view α of the lens

$$\boldsymbol{\alpha = atan\left(\frac{w}{2D}\right)}$$

The range of the angle of view can be defined to cover an area from w=10cm to w=20cm. With this range we have that α must be compressed between 20.48º and 37.43º.

The angle of view determines the relation of the image sensor size d and the focal length f of the optical element with the equation

$$\alpha = 2 \times atan\left(\frac{d}{2f}\right)$$

Before computing the required focal length we must know the image sensor size. Usually image sensors have different width and height. We have to take the most restrictive size, which is the smaller one.

The size of the selected image sensor is 4.6mm x 3.7mm. So we can compute the minimum and maximum focal length to be 5.46mm and 10.24mm respectively.

Another parameter we have to take into account is the distance between the image sensor and the optical element. This parameter is also related with the focal length of the element with equation

$$\frac{1}{f} = \frac{1}{S_1} - \frac{1}{S_2}$$

This equation relates the focal length f, the distance between the optical element and the sensor S_1 and the distance between the lens and the focused object S_2 .

4.2.3. Housing

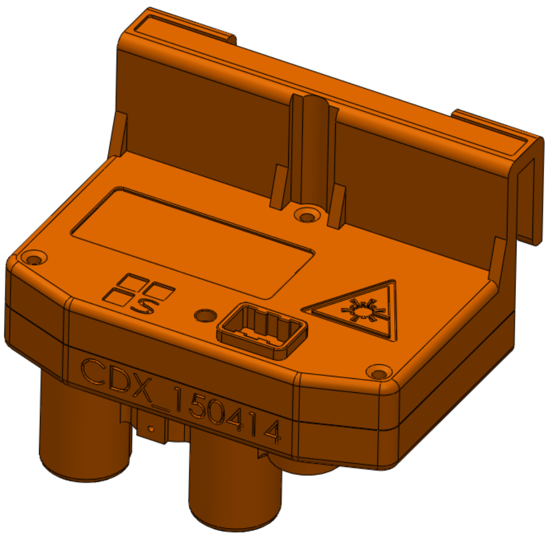


Figure 30. Device Housing 3D

4.3. Software Design and development

In this section we will explain various aspects about the software design and development. The project involves the development of an embedded firmware for the device microcontroller. The embedded firmware not only captures and processes images, but also implement the communication protocol.

We have used ChibiOS/RT to develop the firmware. This is an embedded, open source Real-Time Operating System (RTOS). Using an embedded RTOS allows us to develop faster the firmware. It has mechanisms such as multithreading that facilitates a lot the tasks of firmware development. In the first point of this section, 4.3.1, the architecture of the developed firmware is explained. After that, in subsection 4.3.2 we will explain the image processing algorithms that we have implemented. We have dedicated subsection 4.3.3 to explain the communication protocol. The communication protocol is an important aspect of the development because, as we will see in this subsection, it represents an important challenge. Finally, in subsection 4.3.4 we expose the details of the desktop program that we have developed to configure the device during the installation.

4.3.1. Firmware architecture

The firmware for the microcontroller will be implemented using the embedded real time operating system (RTOS) ChibiOS/RT. We have chosen this RTOS because it is free and open source. Furthermore it have full support for ST microcontrollers and the discussion forum is very active and the questions are answered very quickly.

ChibiOS/RT is multithreading. This fact will allow us to perform some tasks at the same time. In order to pass data among threads there are several mechanisms. There is a message mechanism that allows to send a message to a thread and wait for the response. There is another mechanism, called mailbox, that implements a queue of message. A thread can post a message to a mailbox and another thread can fetch messages from this mailbox. Depending on the functionality we want to implement we will use one or the other.

Figure 31 shows a basic diagram of the threads we have implemented and mailboxes and messages between threads.

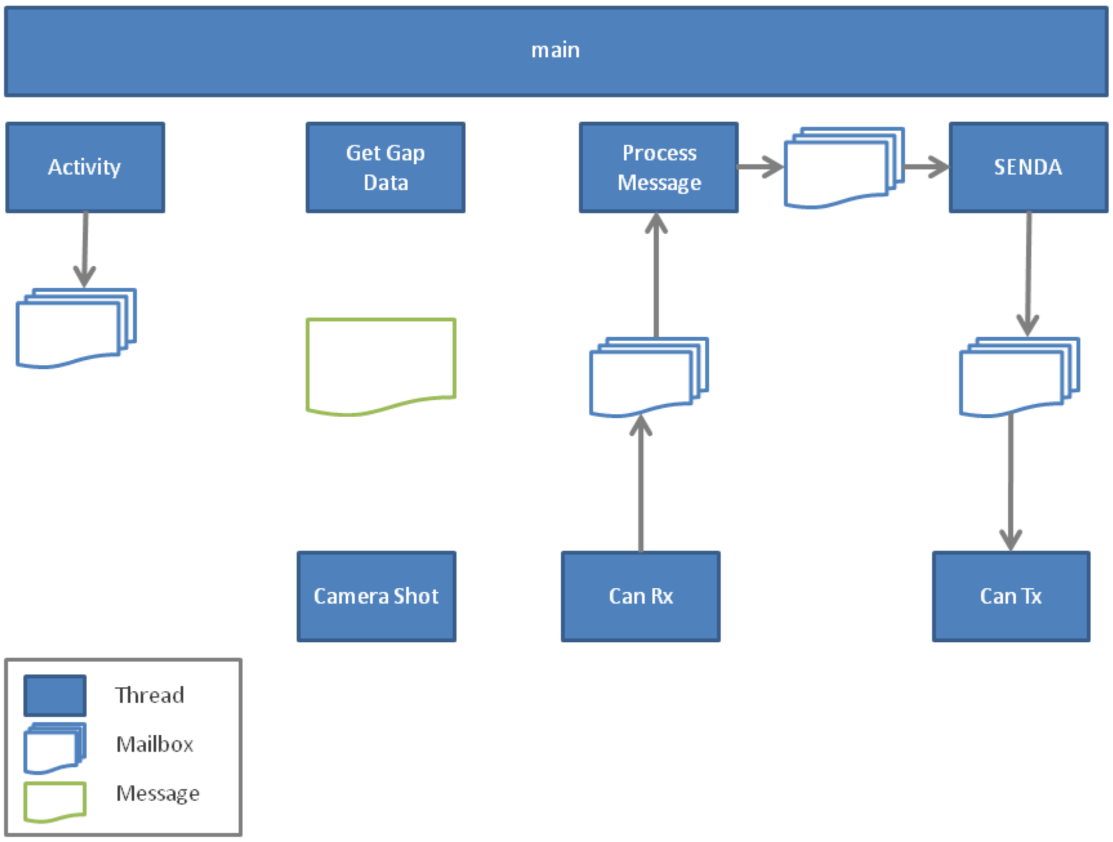


Figure 31. Firmware architecture

The functionality of the threads and mailboxes is the following:

- **Main**: it initializes all drivers, mailboxes and other threads. When all threads are running it enters in an infinite loop.
- **Activity**: Every programmed period of time it sends a CAN Activity Message to the ECON to inform that the device is alive.
- **Get Gap Data**: this thread sends a message to the thread "Camera Shot" and waits for the image. After receiving the image data it estimates the position of the point machine and computes the gap.
- **Camera Shot**: this thread initializes the camera module to capture one frame.

- **Can Rx**: it listens to the CAN interface and receives messages from the ECON and other devices in the bus. Once it has received one data frame it puts a message in a mailbox for the thread "Process Message".
- **Can Tx**: it receive messages from other threads through a mailbox and sends those message through the CAN interface.
- **Process Message**: it receives messages through a Mailbox from the thread "Can Rx" and processes them.
- **SENDA**: this thread implements the custom communication protocol over CAN that will be discussed in Subsection 4.3.3.

4.3.2. Image processing

The device we are developing has to do two tasks. The first one consists on determining the current position of the point machine. If this task is completed successfully then the device has to measure the gap between the lock bar notch and the lock blade. In order to do this tasks we will use an algorithm similar to the one used by Kaiyan et al. in [11]. Figure 32 there is a diagram of the algorithm.

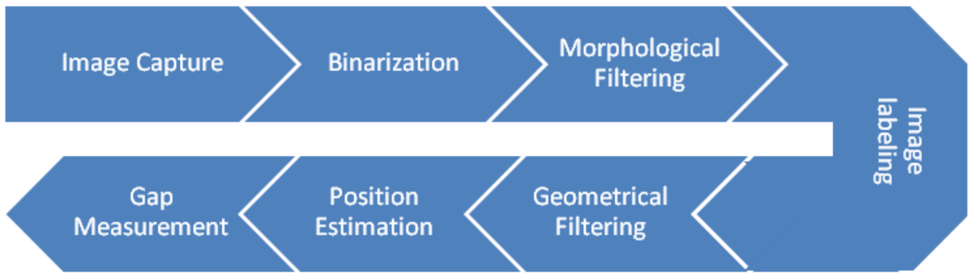


Figure 32. Image algorithm

The first step of the algorithm consists on the image binarization. In order to perform this task we will use the Otsu method which has been explained in Section 3.1. Otsu method gives us the optimum threshold level, all pixels above this level are considered to be the foreground while others are part of the background. After the binarization we apply an opening in order to filter the undesired noise. The most important step in the algorithm consists on extract the connected regions of the image foreground. In order to do so we apply the image labeling method explained in Section 3.3. Once we have found the different connected components in the image we can remove some objects with the geometrical information. We are looking for the laser lines in the image. These are horizontal lines. In the side where the lock blade is present the line will be discontinued while in the other side the horizontal line goes from right to left. In both cases lines we are looking for are connected to the left and/or right margins of the image. Applying this prior information we can remove all the objects that are not connected to the side margins of image. Once we have extracted laser lines we are able to estimate the current position of the point machine. This first task is performed by looking which line is connected to both left and right margins of the image. If we find and object that fulfill this condition we can determine the position of the engine. Once we know the current position we have to measure the existing gap. In order to this task we will focus on the broken line. We will find the endings of both parts of the line in order to establish the number of pixels of the gap. To transform the number of pixels into a real measure we need some reference. We will use the lock blade as the reference because its size is always the same. In order to do this we have to measure this part of the point.

In order to increase the processing speed we have defined a region of interest (ROI). Figure 33 shows the region of interest when the point machine is in normal and reverse position. The ROI consists of 4 windows. The windows are placed in the borders of the locking hammers. In 4.3.2.5 and 4.3.2.6 we will explain how these windows are used to determine the position of the point machine and to compute the gap.

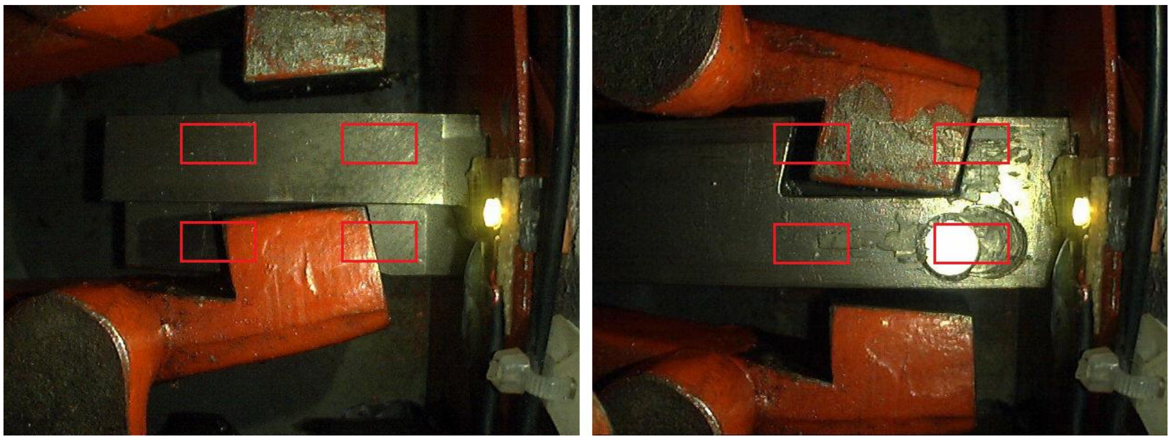


Figure 33. Region of interest in Normal and Reverse position

All the algorithms have been implemented in Matlab before. This allows us to develop the software faster and begin developing the final firmware with tested algorithms. It is very hard to debug image algorithms in the microcontroller. Matlab prototypes help us to compare the results and assess that the firmware is performing correctly.

4.3.2.1. Image binarization

The main step in image binarization consists on finding the optimal threshold to extract objects from the background. In order to perform this step we are using the Otsu's method which has been explained in Section 3.1. Algorithm 1 shows the algorithm steps. The first step consists on computing the image histogram. This is accomplished by initializing an array of length equal to the number of gray levels in the image to 0's. Then for every pixel in the image the counter corresponding to its gray level is increased by 1. At the end of the scanning the counter array contains the number of pixels with each gray level in the image n_i. Then the probability for each gray level is computed as

$$p_i = \frac{n_i}{T}, \qquad i = [0 \cdots L]$$

where $T = M \times N$ is the total number of pixels in the image. It is important to note that

$$\sum_{i=0}^{L} p_i = 1$$

Remember that we are looking for maximizing the intra-class variance of two classes C_0 and C_1. We can compute the first-order momentum of classes as

$$\omega_0(k) = \sum_{i=0}^{k} p_i$$

$$\omega_1(k) = \sum_{i=k+1}^{L} p_i$$

$\omega_1(k)$ can also be more efficiently computed taking into account that the sum of probabilities is equal to 1. Then we can compute it just by subtracting

$$\omega_1(k) = 1 - \omega_0(k)$$

Then at every iteration we only have to add the next p_k to $\omega_0(k-1)$ and compute $\omega_1(k)$ subtracting the previous result to 1.

We can accelerate the algorithm implementation using two conditions. The first one consists on going to the next iteration while $\omega_0(k)$ is equal to 0. This means that there are not any pixels in C_0 and the optimal threshold cannot be in that level because all pixels would be in C_1. The second one consists on ending the loop when $\omega_1(k)$ is equal to 0. This means that there are not any pixels with a gray level greater than the current one so it is not necessary to continue computing.

Algorithm 1. Otsu's method for optimal histogram threshold search

```
1. Compute histogram and probabilities of each intensity level
2. Set up initial $\boldsymbol{\omega_i(0)}$ and $\boldsymbol{\mu_i(0)}, \boldsymbol{i} = [\mathbf{0}, \mathbf{1}]$, and set $\mathbf{max}(\boldsymbol{\sigma_B^2}) = \mathbf{0}$
3. Step through all possible thresholds k=1 ... L
      a. Update $\boldsymbol{\omega_i(k)}$ and $\boldsymbol{\mu_i(k)}$ at every threshold level
      b. If $\boldsymbol{\omega_0(k)} = \mathbf{0}$ go to the next iteration
      c. If $\boldsymbol{\omega_1(k)} = \mathbf{0}$ stop the loop
      d. Compute $\boldsymbol{\sigma_B^2(k)} = \boldsymbol{\omega_0(k)\omega_1(k)(\mu_1(k) - \mu_0(k))^2}$
      e. If $\boldsymbol{\sigma_B^2(k)} > \mathbf{max}(\boldsymbol{\sigma_B^2})$ update it and set $\boldsymbol{k^*} = \boldsymbol{k}$
4. Desired threshold $\boldsymbol{k^*}$ corresponds to the maximium $\boldsymbol{\sigma_B^2(k^*)}$
```

Figure 34 shows an image corresponding to the laser beam, its histogram with the Otsu's level marked with a green line and the binary image extracted with this threshold.

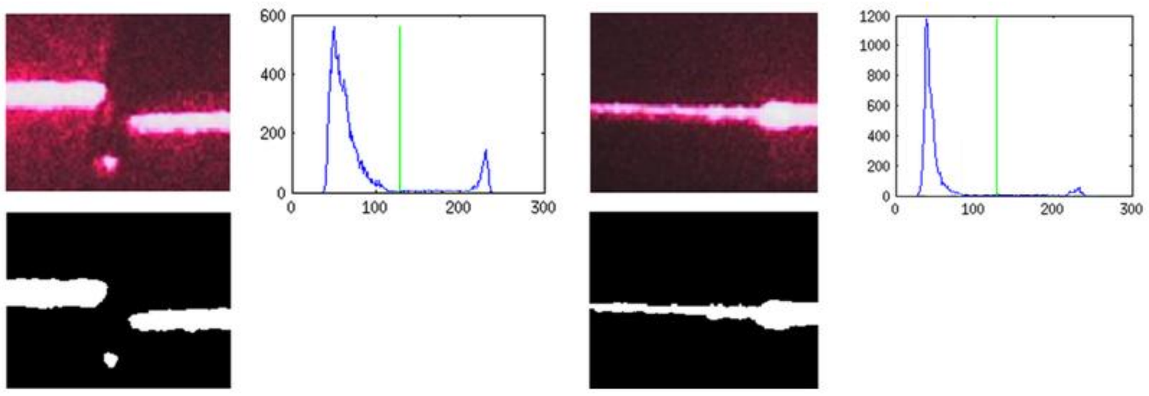


Figure 34. Two examples of laser beam image, histogram and binary image

4.3.2.2. Morphological filtering

We have implemented the basic morphological operators that corresponds to the dilation and the erosion. In order to reduce the number of operations the output image is smaller than the input one. Using this trick we do not need to check whether the structuring element (SE) is completely inside the image or not. For instance, for a square structuring element of size 5, the output image will be 4 pixels smaller in width and height.

Figure 35 in page 44 shows a diagram of the filtering process. In this diagram the structuring element is a square diamond of size 5 with the center of the SE at location 3x3, this is the center of the SE. In the classical approach, we will produce the output pixel (0,0) by placing the center of the SE at location (0,0). This means that there are some pixels of the SE outside the input image and we need to take this into account in some way. When implementing the filtering process we need a loop for placing the SE at every location of the input image. With the classical approach we have to check if the SE is completely inside the image at every iteration. This produces a significant increase in the computational cost of the algorithm.

In the implemented approach we begin by placing the SE at location (2,2) in order to compute the output pixel at location (0,0). Using this approach we do not need to check the position of the SE at every location. We can compensate the size reduction by taking bigger windows to estimate the position of the point machine and the gap measure. As we have said in 4.3.2 all the computations are done inside the region of interest consisting in 4 windows. When cropping this windows we can take into account that some processing steps reduce the image. The initial size of the windows is bigger to compensate this effect. Table 3 and Table 4 shows the code of the implemented functions in C to perform the erosion and the dilation. An opening and a closing has also been implemented by concatenating erosions and dilations as we have explained in the section 3.2, Mathematical Morphology.

Table 3. Implemented erode function

```
void erode(uint32_t width_in, uint32_t height_in, uint8_t *_img_in, uint32_t
*width_out, uint32_t *height_out, uint8_t *_img_out){
  uint8_t se_dim = 2;
  //Pointer conversion to access as a matrix
  uint8_t (*img_in)[width_in] = _img_in;
  uint8_t (*img_out)[width_in-2*se_dim] = _img_out
  int16_t x,y;
  for(y=se_dim;y<height_in-se_dim;y++){
    for(x=se_dim;x<width_in-se_dim;x++){
      if ( img_in[y-2][x]==0  || img_in[y-1][x-1]==0 || img_in[y-1][x]==0 ||
           img_in[y-1][x+1]==0 || img_in[y][x-2]==0  || img_in[y][x-1]==0 ||
           img_in[y][x]==0     || img_in[y][x+1]==0  || img_in[y][x+2]==0 ||
           img_in[y+1][x-1]==0 || img_in[y+1][x]==0  || img_in[y+1][x+1]==0 ||
           img_in[y+2][x]==0)
      {
        img_out[y-se_dim][x-se_dim] = 0;
      } else {
        img_out[y-se_dim][x-se_dim] = 1;
      }
    }
  }
  *width_out = width_in - 2*se_dim;
  *height_out = height_in - 2*se_dim;
}
```

Table 4. Implemented dilate function

```
void dilate(uint32_t width_in, uint32_t height_in, uint8_t *_img_in, uint32_t
*width_out, uint32_t *height_out, uint8_t *_img_out){
  uint8_t se_dim = 2;
  // Pointer conversion to access as a matrix
  uint8_t (*img_in)[width_in] = _img_in;
  uint8_t (*img_out)[width_in-2*se_dim] = _img_out;
  int16_t x,y;
  for(y=se_dim;y<height_in-se_dim;y++){
    for(x=se_dim;x<width_in-se_dim;x++){
      if ( img_in[y-2][x]==1    || img_in[y-1][x-1]==1 || img_in[y-1][x]==1 ||
           img_in[y-1][x+1]==1 || img_in[y][x-2]==1   || img_in[y][x-1]==1 ||
           img_in[y][x]==1     || img_in[y][x+1]==1   || img_in[y][x+2]==1 ||
           img_in[y+1][x-1]==1 || img_in[y+1][x]==1   || img_in[y+1][x+1]==1 ||
           img_in[y+2][x]==1)
      {
        img_out[y-se_dim][x-se_dim] = 1;
      } else {
        img_out[y-se_dim][x-se_dim] = 0;
      }
    }
  }
  *width_out = width_in - 2*se_dim;
  *height_out = height_in - 2*se_dim;
}
```

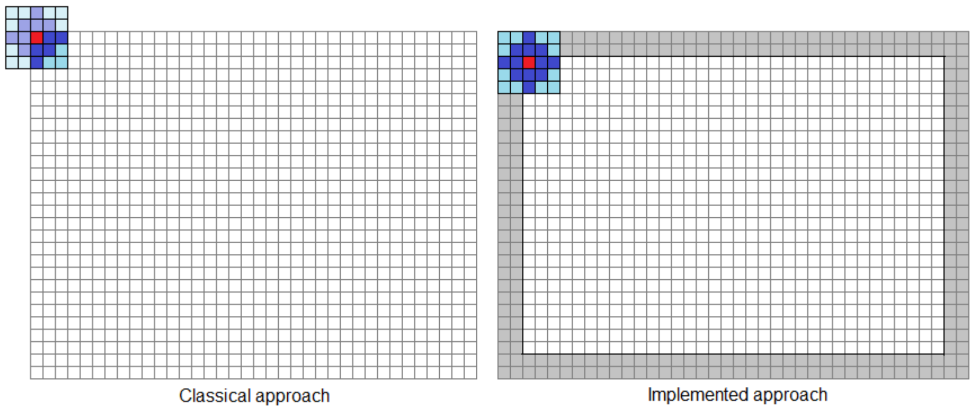


Figure 35. Morphological filtering diagram. Classical and implemented approaches

4.3.2.3. Image labeling

The next step, after filtering the image, is to label the connected components. We want to detect the laser lines. This lines are like an object in our image so we need some mechanism to extract this information. In order to extract objects from an image we have to find the connected components as we have seen in section 3.3 of chapter 3.

We have implemented a two-scan image labeling algorithm similar to the one presented in [19]. A detailed description of the algorithm can be found in Algorithm 2.

Algorithm 2. Two-scan labeling algorithm

```
1. Initialize a correspondence matrix of size MAX_LABELS x 2 to -1
2. Initialize an output labels map of the input image size to -1
3. Initialize a vector of size MAX_LABELS to 0 to store the area of each label
4. Initialize var num_labels = 0
5. For each pixel in the input image except the first and last row and column:
    a. If in(x,y) == in(x-1,y-1) and out(x-1,y-1) has not initial value then assign
       out(x,y) same label as out(x-1,y-1)
    b. If in(x,y) == in(x,y-1) and out(x,y-1) has not initial value then
        i. If out(x,y) has initial value assign out(x,y) same label as out(x,y-1)
        ii. Otherwise insert a correspondence [out(x,y), out(x,y-1)]
    c. If in(x,y) == in(x+1,y-1) and out(x+1,y-1) has not initial then
        i. If out(x,y) has initial value assign out(x,y) same label as out(x+1,y-1)
        ii. Otherwise insert a correspondence [out(x,y), out(x+1,y-1)]
    d. If in(x,y) == in(x-1,y) and out(x-1,y) has not initial then
        i. If out(x,y) has initial value assign out(x,y) same label as out(x-1,y)
        ii. Otherwise insert a correspondence [out(x,y), out(x-1,y)]
    e. If out(x,y) has initial value then assign num_labels to out(x,y) and increase
       num_labels by 1
6. Initialize a vector of size num_labels to store the final label value to -1.
7. For each correspondence as cor in the correspondence matrix:
    a. If label(cor[0]) and label(cor[1]) has initial value then assign both
       label(cor[0]) and label(cor[1]) value cor[0]
    b. Otherwise if label(cor[0]) and label(cor[1]) has not initial value and
       label(cor[0]) is different from label(cor[1]) then reasign label(cor[0]) to all
       labels which has value equal to label(cor[1])
    c. Otherwise if label(cor[0]) has not initial value then assign value of
       label(cor[0]) to label(cor[1])
    d. Otherwise assign value of label(cor[1]) to label(cor[0])
8. For each label in the initial label map substitute the original label by the
   corresponding one and add 1 to the area of such label in the area vector.
```

In the first scan we assign a label to each pixel in the output label map taking into account 8-connectivity. As we scan the image from top to bottom and left to right we only have to check 4 neighbors as we can see in Figure 36. As in the previous step, morphological filtering, we reduce the output image by 1 pixel in order to reduce the computational cost of the algorithm.

🟦	🟦	🟦
🟦	p	-1
-1	-1	-1

Figure 36. Label neighbors

When two labels connected pixels belongs to the same object but has different labels a correspondence is inserted in a correspondences matrix. Once we have concluded the first scan we construct a vector of labels taking into account all the correspondences we have found. In the second scan, for each label in the initial label map we change the label according to this new vector of labels.

In Figure 37 we can see two examples of the labeling process. In the left we can see 3 objects in the foreground while in the right there is only one connected component.

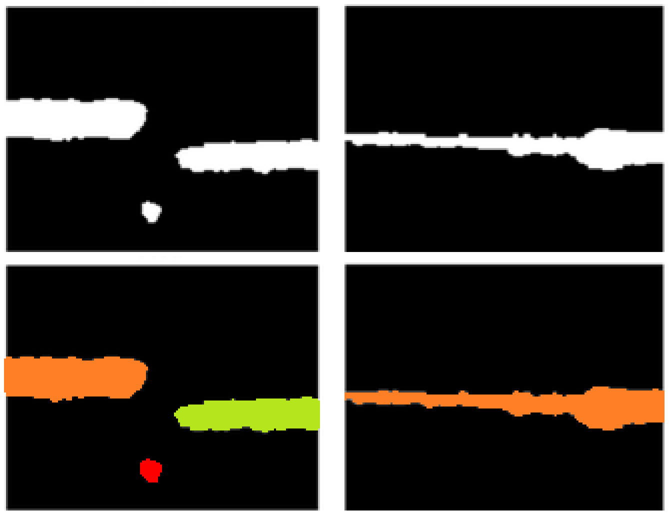


Figure 37. Example of labeling connected components

4.3.2.4. Geometrical filtering

4.3.2.5. Position estimation

4.3.2.6. Gap measurement

4.3.3. Custom communication protocol over CAN

The device we are developing have to be placed inside the point machine engine housing. Furthermore it have to be integrated in the current monitoring system explained in Section 2.3. The easiest way to integrate the new device into the system is using the same communication bus to transfer the computed data and the captured images when necessary. This bus is a two pair CAN bus. The CAN specification can be found in [18]. The most important points are that a CAN data frame can have up to 8 bytes and CAN specification does not provide neither reliable nor ordered delivery of data frames

The transmission of the point machine position estimation and the gap measurement can be done within one single data frame so it does not requires an special protocol. A simple ACK mechanism is used to ensure that the data arrives to its destination.

In the other hand, one data frame it is not enough to send the entire image to the server. We must send a large number of CAN frames. We cannot send an ACK for every single data frame so we must implement a custom protocol over the CAN layer to ensure that all the image bytes arrive to their destination.

The custom protocol must ensure that all frames arrive to the communications concentration device and that they arrive in the right order. Figure 39 shows the diagram of this protocol. The key aspects of the protocol are the following:

- The transmission has an identification number. Every data frame corresponding to the same transmission contains one byte with this identification number.
- The first frame of the protocol contains the number of bytes that will be sent.
- Each data frame with image data bytes contains a sequence number indicating the position of the data.
- ACK are used in order to ensure that all frames arrive to the receiver.
- NACK are send to the transmitter to say that some frames have been lost. The transmitter must send again the lost frames.

4.3.3.1. Protocol description

Figure 38 shows the structure of the data frames with the information that contains each one. The protocol, which can be seen in Figure 39, is as follows.

1. The protocol can start by two different ways:
    a. Some device in the system sends a message (*Send Data Request*) indicating to the device that it has to send an image. In this case, the device sends a message indicating that it has received the request and will take and send the image (*Wait Data*). Then the device takes a picture, performs the corresponding computations and it sends the data and/or the image.
    b. The device has detected an abnormal situation and it decides to send the image.
2. In both cases, when the device has an image ready to send to the server, it sends a data transfer request (*Image Properties*) to the communication concentration device (ECON). In the data transfer request there is information about the type and length of the data. When the ECON is ready to receive the image it sends an *ACK Start Image* to the device. Then the device starts the transmission of the image. The device has a timeout mechanism to send as many times as needed the *ACK Start Image*. If the ECON has not enough memory to store the entire image it sends a message to the device (*Abort Tx*) indicating that it cannot transmit the data. If the image properties do not match the ones that the ECON has previously stored it sends a message indicating that it is receiving another transmission and the device has to wait to send its image (*Wait Tx*).
3. Data frames (*Image Frame*) containing image bytes are grouped in blocks of 40 frames and each frame contains a data frame sequence number. The device will send the first block of 40 frames and then an *ACK End Block*. When the ECON has received the *ACK End Block*, it checks whether it has

correctly received the 40 data frames of the current block. If this condition is fulfilled it sends an *ACK* to the device indicating that it can send the next block of 40 frames. In a noisy environment with many devices connected to the same bus is usual that some data frame gets lost during the transmission. When the ECON receives the *ACK End Block* and there are some data frames missing or when the *ACK End Block* is lost, it sends a *NACK Frames Block*. The *NACK* contains 5 bytes indicating which packets have been lost. Each bit of these bytes indicates with a 1 that the corresponding data frame has been lost. For instance, the code ...00010010 means that frames 2 and 5 have been lost and they need to be sent again.

4. The last ACK End Block sent by the device contains one bit indicating that the transmission is complete. If the ECON has received as many data frames as the first *Image Properties* message has indicated it sends the image to the server through the Ethernet interface. The Ethernet protocol is not described here because it is not within the scope of this project.

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

telecom BCN

ECON TO SOC

Send Data Request		SID	EID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
				1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
	Description		SOCEID	Msg To SOC	Type	Data Type	Id Image				
	Value	SID SLAVE	Device ID	55	170	63	[0-255]				

ACK Start Image		SID	EID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
				1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
	Description		SOCEID	Msg to SOC	Type	Id Image	Image size in frames				
	Value	SID SLAVE	Device ID	55	171	[0-255]	Number of frames				

ACK Frame Data Block		SID	EID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
				1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
	Description		SOCEID	Msg To SOC	Type	Id Image	Block Number				
	Value	SID SLAVE	Device ID	55	175	[0-255]	Block Sequence Number				

NACK Frames Block		SID	EID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
				1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
	Description		SOCEID	Msg To SOC	Type	Id Image	Frame Error Mask				
	Value	SID SLAVE	Device ID	55	172	[0-255]	1 bit for frame (40 frames)				

Abort Image Tx		SID	EID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
				1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
	Description		SOCEID	Msg To SOC	Type	Id Image					
	Value	SID SLAVE	Device ID	55	173	[0-255]					

SOC TO ECON

Image Properties		SID	EID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
				1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
	Description	S_TO_E	SOCEID	Type	ID Image	Image size in frames					
	Value	0x00B	Device ID	70	[0-255]	Number of frames					

Image Frame		SID	EID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
				1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
	Description	S_TO_E	SOCEID	Type	ID Image	Frame #		Image data			
	Value	0x00B	Device ID	71	[0-255]	Dataframe Sequence Number					

ACK END BLOCK		SID	EID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
				1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
	Description	S_TO_E	SOCEID	Type	ID Image	Block Number		End			
	Value	0x00B	Device ID	75	[0-255]	Block Sequence Number					

Abort Data Rx		SID	EID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
				1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
	Description	S_TO_E	SOCEID	Type	ID Image						
	Value	0x00B	Device ID	72	[0-255]						

Wait Data		SID	EID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
				1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
	Description	S_TO_E	SOCEID	Type	data type						
	Value	0x00B	Device ID	73	63						

Figure 38. Image Transmission data frames

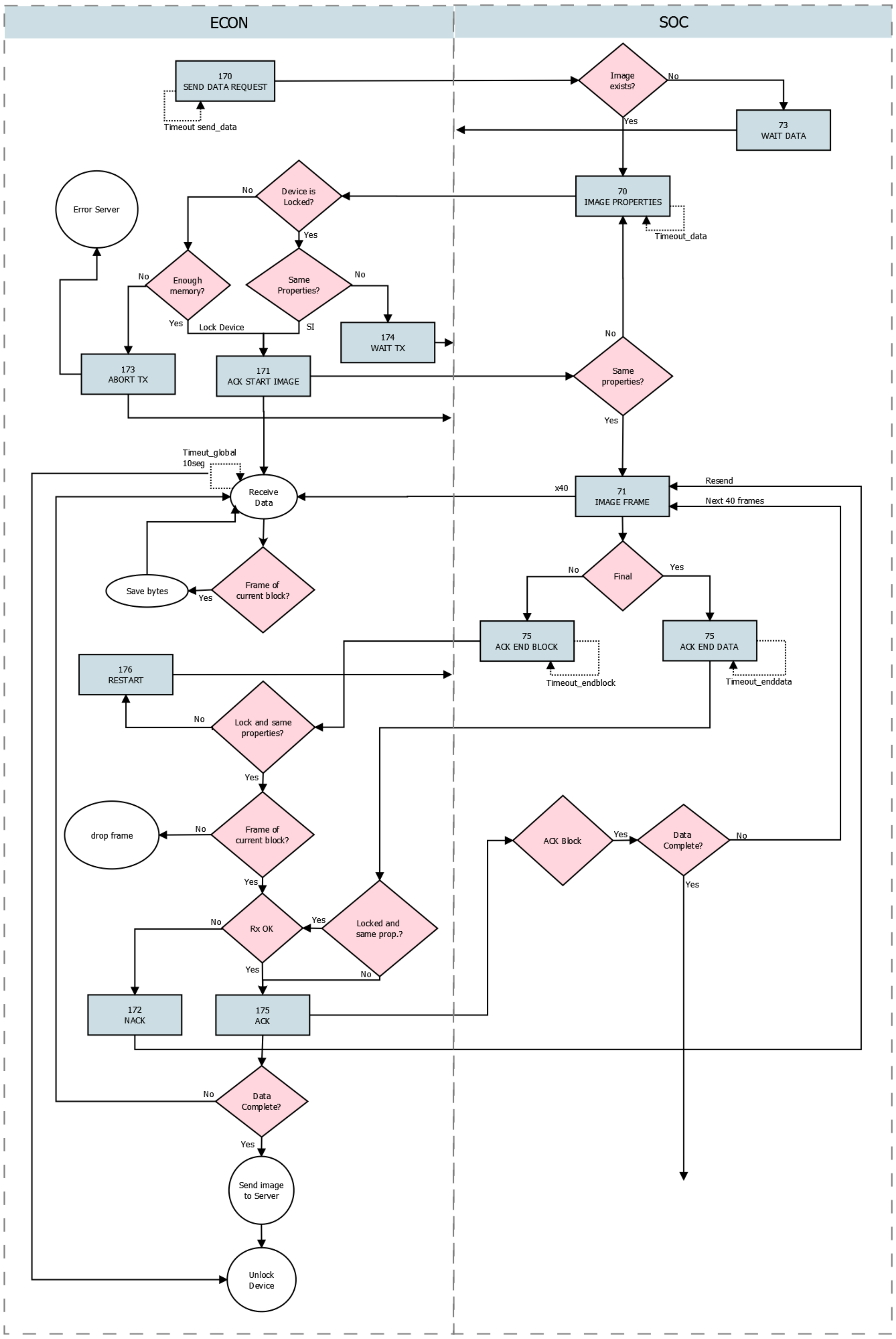


Figure 39. Custom communication protocol diagram

4.3.4. Installation Program

A desktop program has been developed to configure the device during its installation.

5. Results

This should include your data analysis and findings.

6. Budget

Depending on the scope of the thesis, this document should include:

- Components list with approximate costs (prototype).
- Design, prototyping and other tasks costs (hours_person x cost).
- Financial viability analysis.

7. Conclusions and future development

This should include your summary, conclusions and recommendations.

Bibliography

A thorough reference list such as that shown in the following examples: Conference paper [1], journal paper [2], book [3], standard-1 [4], standard-2 [5], online reference [6], patent [7], M.S. thesis [8] and Ph.D. dissertation [9].

[1] R. C. Franke. "Railway Switch Machine Point Detection System". Patent US 6382567 B2. 25 Aug, 1999.

[2] M. A. Hager, M. F. Towey, Jr. "Contactless point detection system for railroad switch". Patent US 6427949 B1. 23 Jan, 2001.

[3] A. Girbau, M. Frigola, M. Gispert, E. Cappellino. "Sistema de predicción de fallos en redes ferroviarias". Patent ES 2374465 B1. 18 Dec, 2009.

[4] M. L. Baird, "SIGHT-I: A Computer Vision System for Automated IC Chip Manufacture". Systems, Man and Cybernetics, IEEE Transactions on . Nov. 1976, pp. 3-7.

[5] J. F. Jarvis. "A Method for Automating the Visual Inspection of Printed Wiring Boards". Pattern Analysis and Machine Intelligence, IEEE Transactions on (Volume:PAMI-2 , Issue: 1). Jan. 1980, pp. 77-82.

[6] A. Anzalone, G. Gugliotta, A. Machi', G. Sardisco. "Automatic Quality Control of Industrial Products for Irrigation". Image Analysis and Processing, 1999. Proceedings. International Conference on. Sep. 1999, pp. 588-593

[7] Y. Rong, D. He, Y. Lin. "Rapid Detection Method for Fabric Defects Based on Machine Vision". Computer Application and System Modeling (ICCASM), 2010 International Conference on (Volume:10). Oct. 2010, pp. 662-666.

[8] C.J.Zhao, G.Q. Jiang. "Baseline detection and matching to vision-based navigation of agricultural robot". Wavelet Analysis and Pattern Recognition (ICWAPR), 2010 International Conference on. July 2010, pp. 44-48.

[9] F. Kunwar, B. Benhabib. "Rendezvous-Guidance Trajectory Planning for Robotic Dynamic Obstacle Avoidance and Interception" Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on (Volume:36 , Issue: 6). Dec. 2006, pp. 1432-1441.

[10] A. Zaki, M. Eskander. "Spray Painting of a General Three-Dimensional Surface". Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on (Volume:3). Oct. 2000, pp. 2172-2177.

[11] L. Kaiyan, W. JunHui, C. Jie, S. Huiping. "Measurement of Plant Leaf Area based on Computer Vision". Measuring Technology and Mechatronics Automation (ICMTMA), 2014 Sixth International Conference on. Jan. 2014, pp. 401-405.

[12] N. Pauly, N.I. Ra_a, "An Automated Embedded Computer Vision System for Object Measurement". Circuits and Systems (MWSCAS), 2013 IEEE 56th International Midwest Symposium on. Aug. 2013, pp. 1108-1111

[13] M. Kamarajui, P.A. Kumar. "DSP based Embedded Fingerprint Recognition System". Hybrid Intelligent Systems (HIS), 2013 13th International Conference on. Dec. 2013, pp. 6-11

[14] K. Zhang, W. Tang, H. Wei, R. Shi. "Study on the Identification System of Car License Plate Based on Imbedded Computer System". Education Technology and Computer (ICETC), 2010 2nd International Conference on (Volume:1). June 2010, pp. 146-149

[15] L. Acasandrei, A.l Barriga. "Embedded Face Detection Implementation_. Biometrics Special Interest Group (BIOSIG), 2013 International Conference of the. Sept. 2013, pp. 1-8

[16] "Electrical resistivity and conductivity". [Online] Available: http:// http://en.wikipedia.org/wiki/Electrical_resistivity_and_conductivity. [Accessed: 23 April 2015]

[17] N. Otsu. "A Threshold Selection Method from Gray-Level Histograms". Systems, Man and Cybernetics, IEEE Transactions on (Volume:9). Jan. 1979, pp. 62-66

[18] R. Bosch GmbH. "CAN Specification version 2.0". *Robert Bosch GmbH*, 1991. [Online] Available: http://www.kvaser.com/software/7330130980914/V1/can2spec.pdf. [Accessed: 7 May 2015].

[19] L. He, Y. Chao, K. Suzuk. "A linear-time two-scan labeling algorithm". Image Processing. IEEE International Conference on (Volume:5). Sep. 2007, pp 241-244.

Appendices (optional)

Appendices may be included in your thesis, but it is not a requirement.

Glossary

A list of all acronyms and what they stand for.