UCLouvain

École polytechnique de Louvain

Insects monitoring using a triggered camera device for existing traps

Author: **Louis HANUT**
Supervisors: **Aaron DOLLAR, Renaud RONSSE**
Reader: **Laurent FRANCIS**
Academic year 2021–2022
Master [120] in Electro-mechanical Engineering

Abstract

Recent studies have highlighted a decline in insect abundance and diversity leading to possible disastrous consequences on the food chain and ecosystems functioning. The need for monitoring technologies to reinforce observations is growing.
In this thesis, a non-invasive, open-source and automated solution for insect imaging has been developed. The imaging process takes pictures from 3 different views (top, below and side) and is automated thanks to an infrared (IR) photoreflective sensor located at the device entrance.
The device was designed to interface a malaise trap but is modular and could be extended to other insect traps.
The power consumption of the device is relatively low ensuring an operating time ranging from 8.12 to 17.56 [h] depending on the number of insect visits. The minimum time between two specimen observations is ≈ 8.4 [s] which corresponds to a decent rate.
In addition, a selection code was implemented for post-processing the set of images captured after the IR sensor trigger. This code (WMSE2) aims at keeping only the image where the insect is the most visible. 100% of correct selection were observed under luminance and noise perturbations (same on all images from the set) ranging from 0 to 80 which is the level of variation measured in real conditions operation. The accuracy drops to $\approx 80\%$ when subjected to higher perturbations.

Acknowledgments

At first, I would like to express my deepest gratitude to my supervisors: Prof. Renaud Ronsse (UClouvain) and Prof. Aaron Dollar (Yale University) for their help and wise advises throughout the year. Prof. Renaud Ronsse was always available for meetings (even across the ocean) to guide my work and thesis writing. I thank Prof. Dollar for accepting me in the Grab Lab during my internship. I appreciated the weekly lab meetings and I keep a very valuable experience from my stay at Yale.
I take this opportunity to thank Prof. Laurent Francis who agreed to be part of my jury and to read my thesis.

I would also like to thank the Grab Lab team for their welcome and support during my internship and stay at Yale. Besides providing me an exceptional insight in the robotics research field, I was really glad to meet them. Many thanks to Adam, Andrew, Gaurav, Hector, Quentin, Sam and Vatsal for the unforgettable experience I lived with them in and outside the lab.
I want to thank Zubin and Jacob (Yale undergraduate students) for their help in the insect monitoring project.
I thank my family and friends in Belgium for their support during the whole project.

I gratefully acknowledge the support of the International Lhoist Berghmans Innovation Chair that made my exchange with Yale University possible.

List of abbreviations

ABS	acrylonitrile butadiene styrene
AC	Alternating current
AI	Artificial intelligence
AIR	active infrared
AM	Air Mass
API	Application Programming Interface
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CSF	contrast sensitive function
CT	Connecticut
DC	Direct current
DNA	Deoxyribonucleic acid
GPIO	General-purpose input/output
GPS	Global Positioning System
HDP	High Density Polyethylene
HVS	Human Visual system
IA	Impact Acoustic
IR	infrared
IRSR	infrared sensor-ring

LDV	Laser Doppler vibrometer
LED	Light Emitting Diode
MCC	Moth Classification and Counting
MPU	Micro-Processing Unit
MSE	Mean Squared Error
MVS	machine vision system
OS	Operating system
PIR	passive infrared
PNG	Portable Network Graphics
PSD	Power Spectral Density
RMS	Root mean square
SSIM	Structural Similarity Index
TOF	Time of flight
USA	United States of America
USB	Universal Serial Bus
UV	Ultraviolet
V4L2	Video for Linux 2
VNC	Virtual Network Computing
WMSE1	Weighted Mean Squared Error 1
WMSE2	Weighted Mean Squared Error 2

Contents

Chapter 1

Introduction

Insect is the most widely spread and diverse form of life on earth. They represent more than half of all described living species [1]. Recent studies have observed a global decline in insect abundance and diversity [2], [3]. This is particularly worrying because it could cause cascading effects in the food chain and ecosystems. Indeed, insects play a crucial role in numbers of processes such as pollination. For instance, 80% of the plants depend on insect pollination [3]. They are also an important food source for higher levels of the food chain (birds, mammals...). The threats associated to insect decline are climate change, intensive agriculture, habitat deterioration and invasive species [2], [3]. Article [2] also identifies that the nitrification of the atmosphere (fossil fuels burning) jeopardizes insect life. The decline magnitude and geographic range are still not quantified and the need of automated monitoring technologies is growing.
In opposition, insects can also have negative effects on agricultural production [4], [5], [6]. Pests reduce crops yield. Monitoring devices allow to deal efficiently with invasion (reducing the pesticide use for instance). Another application where we need insect regulation is for controlling disease transportation.

Manual monitoring techniques are not efficient because an accurate identification requires a high expertise in entomology [5], [7]. The process is also time consuming and costly. An automated solution that performs detection and information extraction is therefore of great interest. This would also enable longer operating periods leading to robust conclusions about insect diversity and abundance over time [8].

The use of trail cameras has gained popularity in wildlife observation recently [9]. These cameras are directly installed in the nature and are paired with a sensor to trigger the imaging process. It results in low power cost for the observation of animals in their habitat. One can wonder how such principle could be transposed to insect monitoring applications.

The previously mentioned concerns on insect populations motivate the conception of an automated, non-invasive and long operating insect monitoring device. The objectives of this thesis are multiple. First, a sensor is selected as a detection sensor that triggers and automates an imaging process to save pictures for future analyses. These analyses could be done by an expert or by an automated identification device. In addition, the conceived device should not harm the observed insects. Lots of current monitoring technologies kill insects to facilitate the identification (e.g. collection in ethanol to preserve Deoxyribonucleic acid (DNA) [8]). A non-invasive solution makes even more sense when observing threatened and scarce species. The goal is to design a monitoring device that takes pictures of living insects and lets them go back to their habitat afterwards. For generality, the designed solution should be versatile and connectable to various existing traps. The trap would ensure to bring living insects towards the conceived imaging device. Finally, to allow non-technical experts to use it, the design should be open-source (easily replicable) and user-friendly.

The chapter 2 of this thesis is a literature review presenting key considerations for the insect monitoring device. This includes an explanation of the working principle of trail cameras to determine what can be transposed to this project, a collection of relevant insect characteristics for monitoring purposes and an overview of the state of the art in automated insect observation.
Chapter 3 covers then the complete conception of the device. On an electrical point of view, it gathers the selection of the different components (detection sensor, cameras...) as well as their characterization and integration with a microcontroller. On the structural side, the assembly of an imaging box and its combination with an existing trap are described.
The next chapter (chapter 4) addresses all the software elements including insect detection, image acquisition and post-processing.
Chapter 5 evaluates the design in real conditions.
Finally, chapter 6 discusses the obtained results and identifies the contributions, limitations and perspectives of this work.

Chapter 2

Literature review

This chapter aims at providing relevant information for the insect monitoring device from the literature. Currently, camera traps are widely used for wildlife observation. The first section is therefore investigating the working principle of those devices to see what can be transferred to this project. Insects being the target of the device, the next section is presenting some of their characteristics that can be used for monitoring purposes. Finally, section 2.3 gathers different state of the art implementations using cameras or not for insect monitoring.

2.1 How does a trail camera work

The current observation of wildlife has been greatly improved thanks to automatic monitoring devices such as trail cameras. Animal motion is important to observe for environmental reasons (detecting invasive species, measuring the impact of environmental change ...). To this end, one can distinguish two different methods. The Lagrangian approach focuses on a single individual using a Global Positioning System (GPS) tag for example. The Eulerian approach records the passage of any species through a fixed place (using a trail camera e.g.) [10]. These cameras are installed in the nature and are paired with a sensor to trigger the imaging process. The detection can be done in many different ways, using a mechanical switch, passive infrared (PIR) or active infrared (AIR) sensors, microwaves detectors or time-lapse triggers [11]. The passive infrared sensors are the most commonly used technology at the moment. They are based on the difference of thermal energy measured in the environment when an animal enters its vision field. Active infrared cameras (AIR), in contrast, uses a pair of infrared emitter/receiver to sense the presence of a body in the field of view. These are often more expensive but more accurate too [9]. Beside the detection element, camera traps are usually constituted of a flash (infrared or not). An IR flash is preferred to reduce the interference with the wildlife natural behavior. The typical structure of a trail camera is shown in figure 2.1.

Depending on the application, the device can be configured to capture images, short videos or time-lapses. In this section, I will present the theory of thermal radiation and

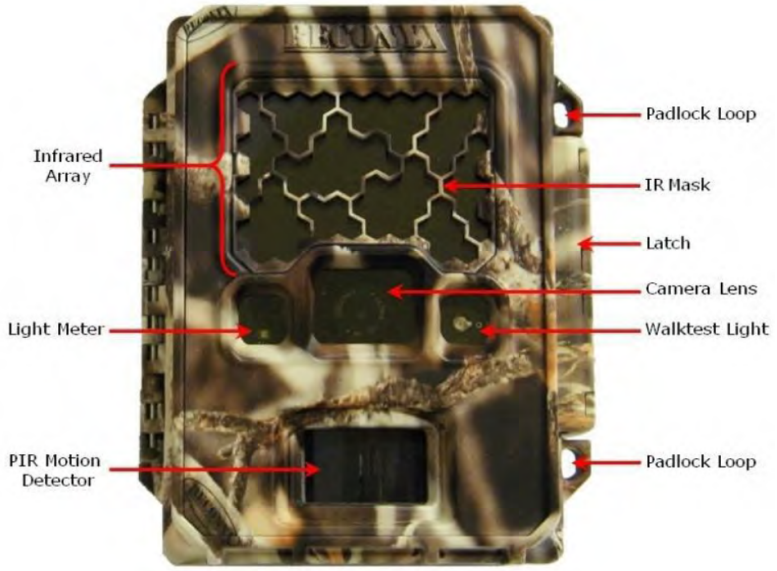


Figure 2.1: Diagram of a Reconyx HC600 showing the various components that are fairly standard in all camera traps (courtesy of Reconyx) [9]

its application to animals and going deeper in the sensing device used in most of the current camera traps.

Thermal radiation

To better understand the underlying principle used in the most widely spread detection sensor (PIR), it is important to discuss the thermal radiations emitted by a body. In 1860, Gustav Kirchoff introduced the concept of blackbody to describe this phenomenon [12]. A blackbody is a perfect emitter, it absorbs all incoming radiations and reflects none. It emits radiations as a function of its temperature. Those radiations are usually not in the visible spectrum leading the body to appear 'black' from which this concept is named. The radiations are not uniformly distributed along the different wavelengths. This distribution is given by Planck's radiation law [13]:

$$F(\lambda) = \frac{2\pi hc^2}{\lambda^5 \left(\exp\left(\frac{hc}{k\lambda T}\right) - 1\right)} \tag{2.1}$$

where F is the spectral irradiance $[W/m^2]$, λ is the wavelength $[m]$ and T is the blackbody temperature. $h = 6,626.10^{-34}$ $[J\ s]$ and $k = 1,380649.10^{-23}$ $[J\ K^{-1}]$ are the Planck's

and Boltzmann's constants respectively. The curves obtain with Planck's radiation law at different temperatures are shown in figure 2.2. The curve in black corresponds to the classical theory, also known as Rayleigh–Jeans law. This theory presented infinite irradiance at low wavelengths which is impossible. When observing the graph in figure 2.2, it may be seen that, at lower temperatures, the peak irradiance is shifted to higher wavelengths. The concept of blackbody is ideal and most of the objects, animals, plants

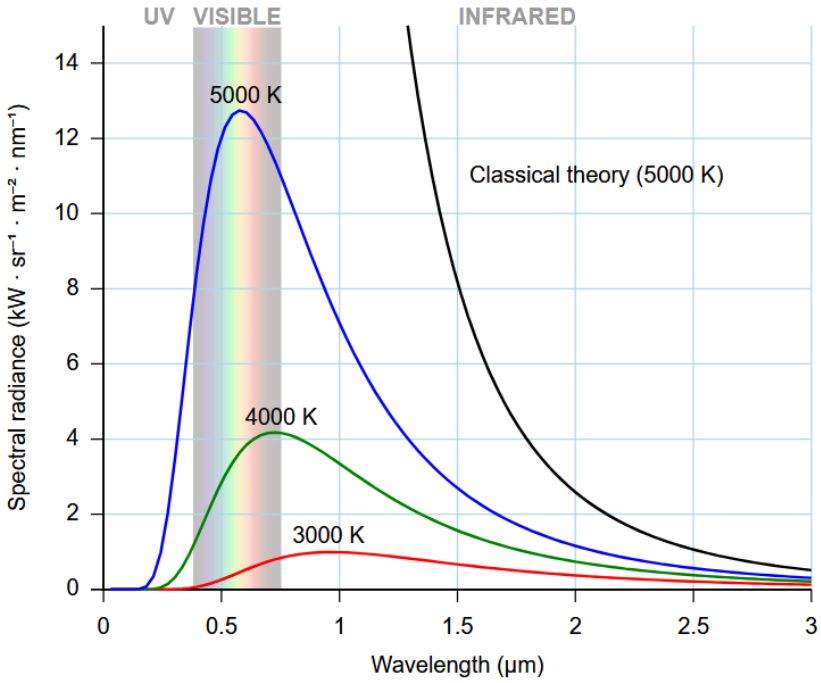


Figure 2.2: Blackbody curves of Planck for various temperatures [12]

and humans are not perfectly described by this theory. The deviation of a given body from an ideal blackbody at the same temperature can be characterized by its surface properties: its transmissivity (τ), reflectivity (ρ) and emissivity (ε). For a blackbody, we have $\varepsilon = 1$ since it is a perfect emitter, $\tau = 0$, it lets no radiations go through it and $\rho = 0$ because it also reflects none. Animals and plants differ from blackbodies because their metabolic activity generates heat internally. Although, most of them can be considered as "greybodies", meaning that their emissivity is close to 1 ($\varepsilon > 0.9$) [11].
From Planck's law (eq. 2.1), the thermal emissions from a body at a temperature between 0 and 60°C present a peak around $8 - 11\ \mu m$ approximately. To cover the range corresponding to real environmental conditions, most of the PIR sensors used in wildlife observation applications are designed to detect wavelengths ranging from 8 to 14 μm.
Finally, thermal radiations can be affected by the propagation path. Unlike in vacuum, the molecules present in the atmosphere absorb some wavelengths radiated hence reducing the energy transmitted [11]. It is happening even more for longer propagation paths.

From the physical medium to electrical quantities : the passive infrared detector (PIR)

Knowing that wildlife emits thermal radiations, one may want to measure or detect those to trigger a camera installed in the nature for example. This can be done using a PIR detector. Such a sensor is usually joined with a focusing element and a threshold comparator. This structure is shown in figure 2.3 for a motion detection application. Each of these elements are presented hereafter.

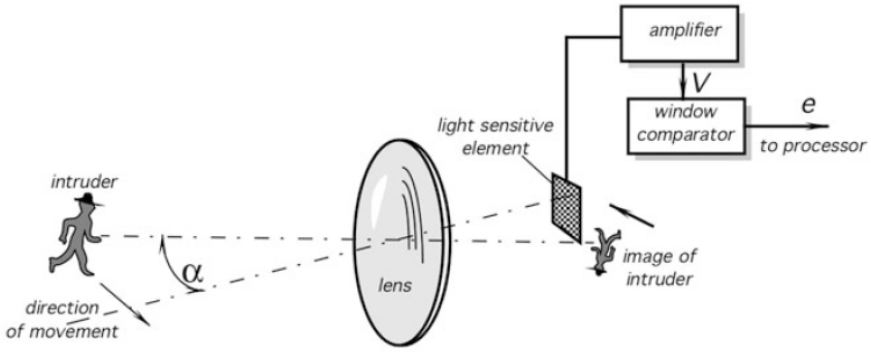


Figure 2.3: General arrangement of optoelectronic motion detector. Lens forms image of moving object (intruder). When image crosses sensor's optical axis it coincides with light-sensing element [14]

Focusing element

The aim of this element is to reproduce the image from the real world on the focal plane of the light-sensing element [14]. To enlarge the field of view, one can use a Fresnel lens as focusing element for example. A Fresnel lens is composed of several smaller lenses (called "facets") to superpose different parts of the environment on the focal plane (see figure 2.6).

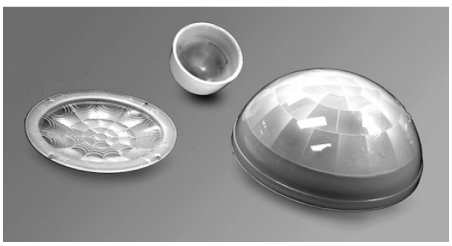


Figure 2.4: Various infrared faceted Fresnel lenses molded of High Density Polyethylene (HDP) [14]

Light-detecting element

As indicated by its name, a PIR is a passive sensor. It is sensitive to thermal radiations emanated by bodies in its field of view but does not emit any radiation itself. Alternatively, active sensors can be used for motion detection. In this case, the device measures the electromagnetic radiations reflected by the target when illuminated by an external source (an infrared LED e.g.).
The light-detecting element in a PIR is most often a pyroelectrics. It is generally sensitive to wavelengths ranging from 4 to 20 μm which corresponds to natural emissions at temperatures between 26 and 37 $°C$ [14]. Pyroelectrics are governed by the pyroelectric effect, i.e. electrical charges are generated in the material in function of the thermal energy flowing through it. More precisely, the pyroelectric effect can be related to the piezoelectric effect (electrical charges created as response to a mechanical stress). When a pyroelectric sensor is subject to a thermal flow, the temperature of the face exposed will increase above the one on the opposite side. This lead to the thermal expansion of the illuminated face and therefore a mechanical stress in the material. The material is also a piezoelectric and electrical charges appear. Connecting electrodes on both sides of the material enable to measure the charges created and thus to detect thermal flow (e.g. the infrared radiations emitted by a body). The pyroelectric effect is graphically shown in figure 2.5. The downside of this technology is that a pyroelectric material could

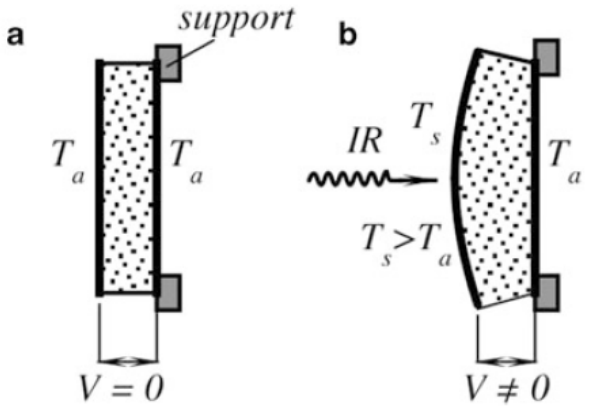


Figure 2.5: Simplified model of pyroelectric effect as secondary effect of piezoelectricity. Initially, the element has uniform temperature (a); upon exposure to thermal radiation, its front side warms up and expands, causing stress-induced voltage across electrodes (b) [14]

be exposed to other mechanical stresses (vibrations, external forces ...) than the one resulting from thermal expansion resulting in false detections. Fortunately, there is a way to get rid of these perturbations using a differential PIR detector. This can be done by

connecting two similar pyroelectric elements such that the mechanical perturbation (that is equal on both sensors) will be canceled. The IR-induced signal will not be affected by this connection since it impacts only one of the sensing elements (the focusing lens is mapping the incoming radiation on a single pyroelectrics).

Threshold comparator

This is the last component of the sensing chain of a PIR sensor. It simply consists of comparing the voltage output from the light sensitive element with a higher and lower threshold (window comparator) to obtain a binary decision : detection or not. When the voltage is outside the window defined by the thresholds, the comparator circuit outputs a logical "high" value meaning motion was detected.

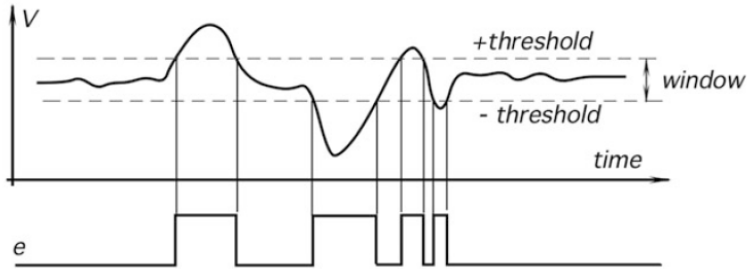


Figure 2.6: Sensor responds with electric signal that is amplified and compared with window thresholds in comparator [14]

In wildlife observation applications, the goal is to capture images from animals in the nature. The background is generally a set of trees, bushes... This is thermally heterogeneous and can sometimes lead to false triggers for the PIR sensor [11].
This concludes the first section that presented how wildlife is currently mostly observed. This also raises an interesting question: can we apply this concept to an insect monitoring task? Or what could be modified to have a working observation device for insects? With this in mind, the next section is focusing on insects and especially the characteristics that could benefit their detection.

2.2 Target specifications for insect traps

Insect is the most widely spread and diverse form of life on earth, they represent more than 50% of all the known species living on this planet. There are many different species and their morphology are varying a lot. Flying or not, from a few millimeters to tens of centimeters, it is important to have a deep understanding of their physical characteristics since it will obviously impact a lot the design of our device.

Section 2.1 showed the widest spread technology in terms of wildlife observation. It is therefore appealing to think about extending it to insects monitoring. An important related question is to determine whether insects are emitting thermal radiations that could be detected. In mammals, the internal body temperature is kept constant in most cases. This is not the case with insects, their temperature changes with their needs. This major difference comes from their relatively small body size [15]. Insects can heat up to ensure a good muscle function for some tasks (e.g. flying) and cool down afterwards. They have developed strategies to do so: insects can shiver or bask (adapt their body posture to capture maximum heat from the surroundings, e.g. a butterfly opening his wings to capture sunshine). The ability of insects to thermoregulate is crucial for them to perform different tasks in various environmental conditions.
Some insects can be characterized acoustically. They produce sounds in various ways depending on their species: by stridulation (produced by friction between two body parts, e.g crickets), percussion (striking a body part, e.g cockroaches), vibration (fast motion of wings in the air, e.g. bees or mosquitoes), tymbal mechanism (contraction of the so-called tymbal muscles, e.g. cicadas) or air expulsion (e.g. cockroaches) [5]. For low intensity sounds, the use of a filter to remove the environmental noise is crucial. Sometimes, when insects can not be directly visually observed, acoustic monitoring methods turn out to be very useful. This is the case in agricultural storage for example, such methods are used to detect and prevent the presence of insect invasions [6].
After this brief overview of insects, it appears that their morphology and behavior vary in many different ways. It is therefore complicated to have one single way to capture insects. Figure 2.7 shows various existing insect traps. The species targeted for each of these are gathered in the table in figure 2.8. Ideally, we would like to propose a device adaptable to those traps to be able to observe as much species as possible.

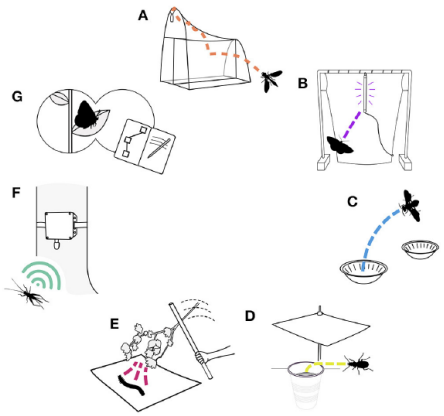


Figure 2.7: A visual overview of the seven insect benchmarking methods summarized here: **(A)** malaise trapping, **(B)** light trapping, **(C)** pan trapping, **(D)** pitfall trapping, **(E)** beating sheet, **(F)** acoustic monitoring, and **(G)** active visual surveys. [8]

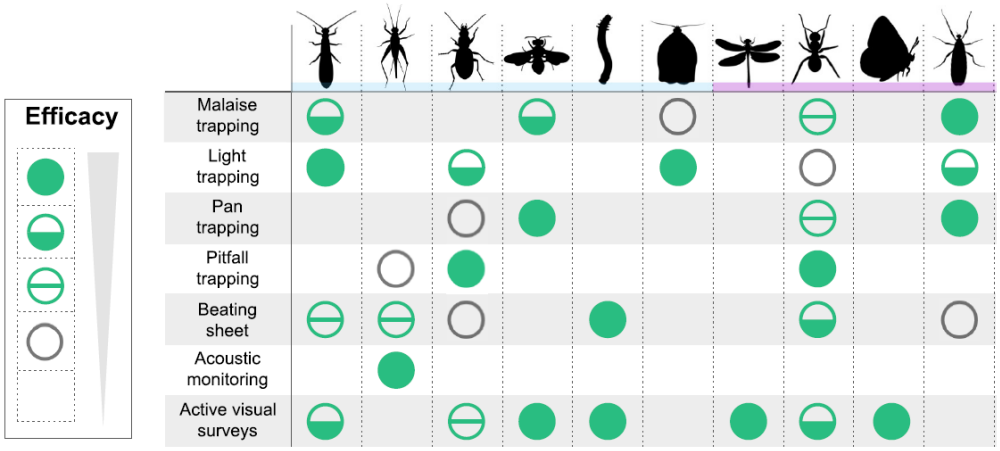


Figure 2.8: Commonly-monitored insect guilds and taxa and the efficacy for each of seven benchmarking methods. Efficacy of each method for a given insect group is scored as follows: filled green circles indicate optimal suitability; half-filled circles indicate possible suitability; divided, unfilled circles indicate marginal suitability; unfilled gray circles indicate bycatch only; and no circle indicates general unsuitability. [8]

Malaise trap

Our design will be initially conjugated with a malaise trap whose targets are flying insects as shown in figure 2.8 (bees, beetles, moths ...). A malaise trap looks like a tent. Insects are hitting a vertical net and have then the tendency to fly upwards to escape the trap. The structure of the nets are such that the insects are then converging towards a collecting vial.
There are different factors to consider when building and installing a malaise trap [8]. First, the design is important to ensure a good capture rate. The vertical net should ideally be black to reduce its visibility and improve the efficacy of the trap. The upwards part of netting is usually white to increase the tendency of flying upwards for trapped insects. In the existing traps, for the collecting container, it is recommended to fill it with a 95 % ethanol solution to preserve the DNA of the collected insects for further analysis. For this work, we want to conceive an automatic non-invasive trap. In other words, once pictures of the insects have been taken, they are free to fly away. Besides, the mesh size of the netting should be accounted for depending on the size of insects targeted. On the other hand, the location of the trap is influencing a lot its efficiency. It is recommended to install it in a natural flight corridor, putting the vertical interception plane perpendicular to the motion direction. Figure 2.9 shows an installed Townes-type malaise trap which is the most commonly used type of trap.

Figure 2.9: A Townes-style Malaise trap in position [16]

2.3 State of the art

The aim of this section is to provide the reader with an overview of the existing implementation of insect traps. Therefore, there will be first a presentation of automated traps and the sensors used to detect the presence of insects. Traps using cameras to characterize insects are then presented. The combination of both researches will be useful for the actual design of this thesis.

Automated insect monitoring devices

In [17], a monitoring system for soil microarthropods is presented. It consists of a network of probes that counts and estimates the size of the small-size insects falling into it. The sensor used to this end is optical and measures the difference in infrared light intensity. This probe is pictured in figure 2.10, the sensing system is illustrated in (c). It is low power-consuming and can work without recharging for 2 to 3 months. As regards detectability, EDAPHOLOG showed 85 % efficiency for the smallest microarthropods. For insects of 1mm long (whatever their morphology), the detectability rises to 90-95 % and to 100 % for larger ones. Some false detection were observed and could possibly be interpreted as soil falling accidentally into the trap.
Though EDAPHOLOG proved to be efficient, it is restricted to the monitoring of small arthropods living in the soil. In [18], this idea is extended to detect insects of different sizes and it provides a detection sensor adaptable to several types of traps. This takes the form of a ring of infrared photo-electric elements (LEDs as light sources and photodiodes as receivers). The use of infrared light is suitable to reduce the perturbation of environmental light. An insect passing through the ring is decreasing the light intensity received by the photodiodes and can then be detected. This device has been tested on different traps and on different species of insects of different morphologies. Two versions

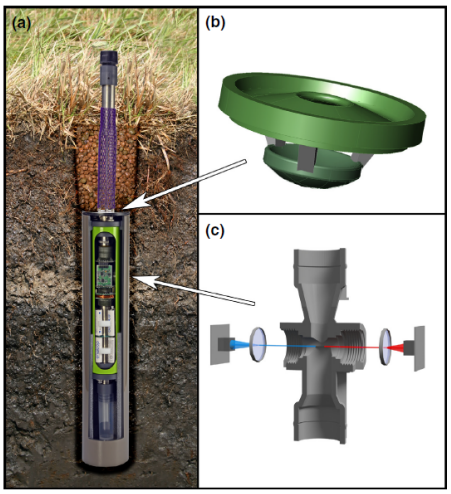


Figure 2.10: EDAPHOLOG probe. (a) The sampling technique is similar to the methodology applied in pitfall traps, but the device catches both soil dwelling and surface active arthropods. Organisms entering the trap through the purple mesh tube are detected upon falling down and preserved in the bottom container. The plastic sieve around the brown clay balls prevents soil particles entering the probe. (b) A small pot within the inner tube impedes entering soil particles to the sensing field. Trapped microarthropods move horizontally, while soil particles remain on the upper plate. (c) Schematic view of the opto-electronic sensor that counts microarthropods and estimates their body size [17]

of the infrared sensor-ring (IRSR) were designed. One for traps targetting large sizes of insects (from 2 to 35 mm). This was achieved by a lower density of infrared detection beams (see figure 2.11(C)). The second version of the sensor ring is rather focusing on microarthropods (from 0.1 to 2 mm) and was tested on a new EDAPHOLOG trap. By placing the detection field closer to the light source and put the photodiodes closer to each other, the sensitivity is increased (see figure 2.11(D)). Figure 2.11(A) shows the whole infrared sensor ring, the mechanical frame is common for both versions.
On the electronic side of the design, there are 4 distinct parts constituting the device : the power supply, the amplifier, the Central Processing Unit (CPU) unit and the communication system. The amplifier is constituted of two consecutive operational amplifiers providing gains of 15x and 30x respectively. The analog output of this amplification stage presents a low frequency component : environmental noise. Using a hard threshold directly on the amplified voltage could result in false detections, a filtering is then necessary within the CPU. To get rid off this unwanted component, a moving average technique is adopted. If the difference between the actual sensing and the computed moving average is exceeding the detection threshold, then we have detection. To assess the chosen detection threshold, experiments has been conducted with particles (grains) and living arthropods of different sizes. A size of particles/arthropods could

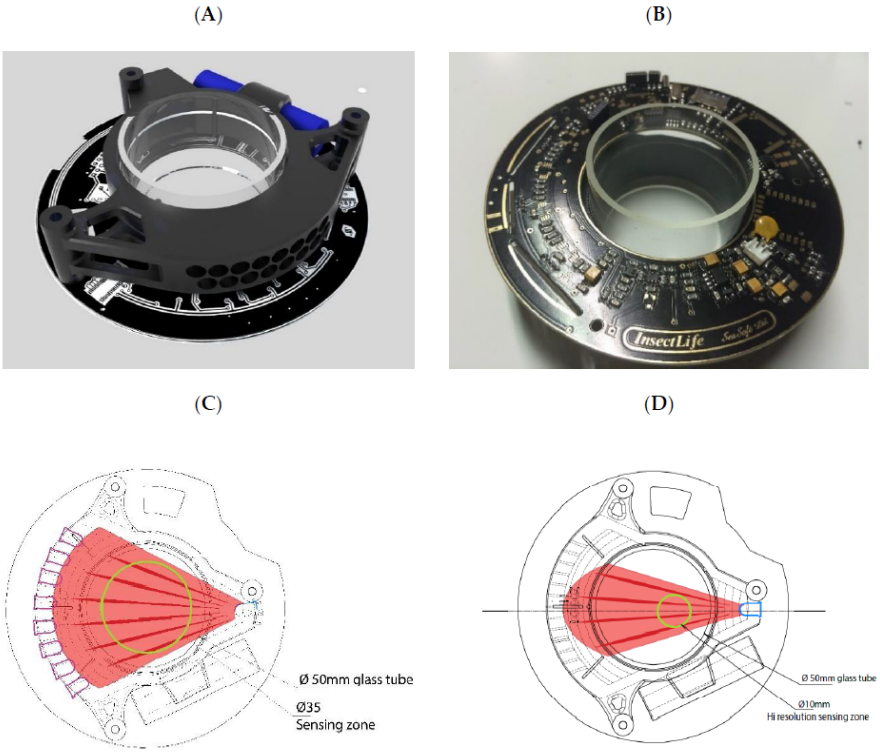


Figure 2.11: The mechanical construction of the infrared opto-electronic sensor-ring (IRSR). (A): whole sensor unit. (B): printed circuit board with a glass tube. (C): the draft of IRSR-1 for larger arthropods. (D): the draft of IRSR-2 for microarthropods. [18]

be associated to a 95% detectability level. For example, in the tests where particles were used, the IRSR-1 presented a 95% detectability level for grains of size 0.9mm. This means that more than 95% of the particles bigger than 0.9mm where detected. The detection reach 100% when the size of particles increase as expected. Beside those statistics, it was observed that detection of living insects was influenced by their morphology.

Those two devices are interesting for the detection of insects of various sizes. Their aim is simply counting the insects in a given area. It could be extended to our project to take images once detected.

In [7], a sensing device made of IR emitters and receivers is tested on different traps. As for the previous technologies presented, the detection of insects is associated with an obstruction of an infrared light beam. Here, the photo-electric elements are placed along 1D arrays to completely cover the entrance of the trap. The emitter is sending pulsed IR signal synchronized with the receiver by a microprocessor unit. The pulses are periodic, have short duration and high amplitude. This Micro-Processing Unit (MPU) is checking

the received pulses with an envelope detector (keeping the amplitude only). The use of pulsed infrared signal ensures a lower power consumption and therefore a longer operating life [19]. The traps in [7] also use pheromones to lure insects inside. Although the batteries can operate for months, the replacement of these attractants requires more frequent human visit of the traps (about twice a month). It is important to note that the pheromones are species-dependent. Experiments to automatically detect adult beetles were then conducted. The accuracy of the device was established by comparing the automatic count with a manual count of the insects entering the different traps. The prototype reached an accuracy of 98-99 %.
Some of the presented technologies were equipped with a communication system. This is interesting to have a remote real time view on the device. Other recorded parameters such as environmental data (temperature, humidity ...) can be sent on top of the counting related data [7].
There are other ways to detect the presence of insects. For agricultural applications, the need to monitor the crops and storage is high. Pests are responsible for losses in production and devices to detect them rapidly and precisely is therefore of great importance. For this reason, acoustic and machine vision system (MVS) methods have been widely developed. A review of sensing technologies to detect insects in crops is presented in [4]. A first possibility is to use accelerometers. When fixed to a plant for example, it can distinguish the vibrations caused by insects. Piezoelectric sensors are also capable to output electric signals when subjected to acoustic vibrations. Those two types of sensors can also be integrated into an acoustic probe that provides more effective detection in different types of media (soil, wood, stored grains ...). Microphones shows promising results with a method called Impact Acoustic (IA) measurement. It is based on the interaction (impact sounds) of samples with a plate (in steel e.g.) and is often used to verify the quality of grain kernels. In another range of the spectrum, ultrasonic transducers can be used to detect insects in wood. This method is based on Time of flight (TOF) measurements, i.e. the time needed for an acoustic signal to travel across some material. One can infer about the presence of insects inside wood samples based on the defects inside. Most of those techniques are contact based, meaning that they need to be mounted on the observed sample (plants e.g.). A non-contact method avoids interference between sensors and specimens and is harmless for the observed environment. Laser Doppler vibrometer (LDV) measures the vibration of a surface without contact. Optical measurements offer the advantage of discarding background acoustic noise compared to the previous methods already explained. Those can be used for flying insect monitoring by analyzing their wingbeat. More precisely, we can measure the way their wingbeat modulates the flow of light. This information can even determine the species of the insect entering the trap. In [19], it is observed that the frequency spectrum is characterized by a high near-Direct current (DC) component (corresponding to the body movement) and higher frequencies harmonics related to the wingbeat itself.

By analyzing the whole spectrum, it is feasible to classify species.
Finally, Machine vision systems offer possibilities to identify insects. It includes visible light, hyperspectral or X-ray imaging sensors. They are precious to extract features from observed insects and then perform classification. Though visible light systems are preferred to extract some features (color, texture, shape...), hyperspectral devices enables to detect specimens when they are invisible to human eye (e.g. in stored grains).

Camera traps for insect observation

In this subsection, I present different articles that are using cameras in automated traps for the monitoring of insects.
In [20], a camera is used to estimate the abundance of nocturnal insects in a non-invasive way. The records then enable to draw conclusions on the spatial and temporal parameters that are influencing this abundance. For this experiment, a camera was oriented towards the sky in different surroundings and was configured to take a picture with flash every 5 minutes. Insects were then counted on each pictures by an observer. A model to predict insect abundance was fitted using the different observations. The relevant features selected were the time, time after sunset, wind, temperature, humidity and habitat. The required parameters could be obtained in real time from a nearby weather station. From the experiments it appears that the greatest insect abundance was obtained for the lake habitat (then wetlands along rivers, forest and finally open area). On the temporal point of view, the period of June/July presents the highest swarming probability. On a shorter scale, the most favorable time for swarm events to occur was just after sunset or right before sunrise. This method is not suitable for species identification but offers a large scale counting possibility. Environmental parameters such as wind, temperature or humidity was influencing the insect abundance differently in the different surroundings. The ambient temperature was not impacting near the lake while an increasing temperature had positive effect on the insect abundance in the other habitats. Wind was decreasing the abundance mainly in the lakes and village environments. A drawback of the presented method is the need of manual counting. The use of an automated identification code based on the photos would greatly reduced the amount of time required for this process.
An automated camera trap specific to moths is presented in [21]. Generally, moths can be observed using light traps. The proposed implementation is using the structure shown in figure 2.12. A sequence of images is captured once the computer vision system is triggered by a motion in the field of view. The purpose of the Ultraviolet (UV) light is to attract from far distances. Vaporizing sugar water on the light table has proved to increase the efficiency of the trap by attracting more insects. The authors developed the algorithm Moth Classification and Counting (MCC) for the computer vision. The first step of it consist of a blob detection, i.e. creating a box around each of the insects of the picture. Each insect is assigned an ID. The ID is then used to track the insect in subsequent images to avoid double-counting. The classification between different

known moth species is handled by a trained Convolutional Neural Network (CNN). The algorithm is summarized graphically in figure 2.13. The network presented promising results : a validation F1-score of 0.93. The F1-score for tracking and classification was in average 0.71. The major penalty of this score was the non-negligible presence of unknown species which were then obviously misclassified.
As this trap is using a computer vision system to detect insects, it requires a continuous operation which is power consuming. This trap is turned off during daytime since moths are nocturnal insects.

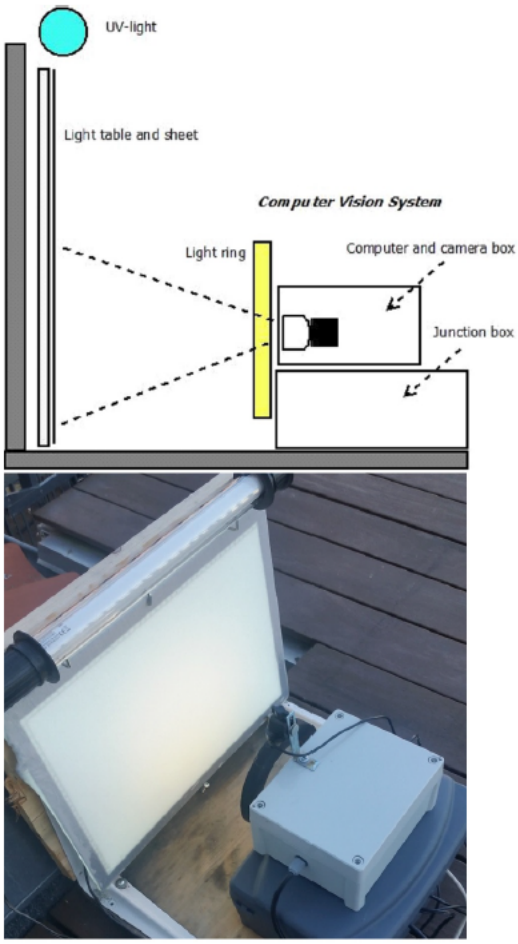


Figure 2.12: The portable light trap with a light table, a white sheet and UV light to attract live moths during night hours. The computer vision system consisted of a light ring, a camera with a computer and electronics, and a powered junction box with DC-DC converter. [21]

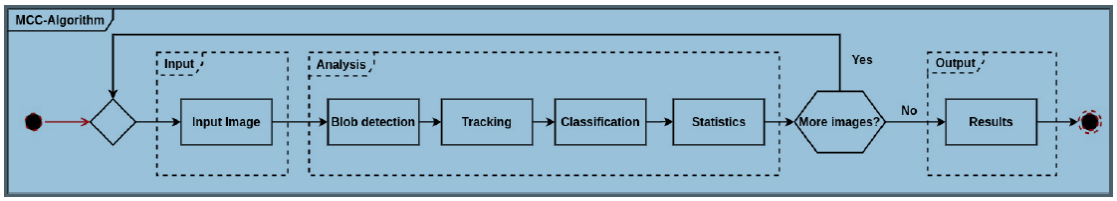


Figure 2.13: The processing pipeline of the intermediate- to high-level image processing algorithms to track and count the number of insects. A customized trained CNN was used for the moth species classification. [21]

Deep learning and entomology

In the two previous subsections, implementations were presented to count specimens and capture images automatically. Deep learning methods can be used to translate the data collected into ecological statistics [22]. Classification models enable to evaluate the abundance and diversity of insects in different ecosystems. A network for moth species classification was already presented in the previous section [21]. Former master theses from UCLouvain were also investigating insect identification using deep learning tools. In [23], an autonomous monitoring trap was designed for nocturnal insects. The insects are attracted with a light source then images are taken and insect recognition is directly performed. A CNN is used for automatic insect identification in [24]. The classification focuses on 10 different classes of Hymenoptera. The Nature Identification Application Programming Interface (API) ([25]) is a web platform for insect and plant identification based on images using Artificial intelligence (AI). Deep learning tools can also provide information regarding insect behavior and species interactions [22]. Cameras allow monitoring of insect continuously and possibly over changing seasons. This can lead to interaction observations such as flower visits by pollinators, defoliation or predation. Finally, [22] highlighted different main directions for future work in the field. First, the validation of species identification using images. Then, the collection of sufficient data for training the identification models. A public database should also be developed to facilitate progress.

Chapter 3

Architecture of the proposed device

This chapter covers the design of the insect monitoring device. It starts with a list of specifications (section 3.1). Section 3.2 deals then with the electrical part of the design, from the selection of the detection sensor to its interfacing with a microcontroller. The next section (section 3.3) describes the structural aspect of the project, explaining how all components are organized together and how the designed imaging device can be integrated with an existing insect trap. Finally, section 3.4 sums up the costs associated to the construction and required purchases involved in this project.

3.1 Specifications

Before interesting to the details of the design, here is a list of specifications that will guide the imaging device conception:

- The monitoring device should be non-invasive. Beside not killing the observed insects, the device should not perturb the environment where it is installed.
- The complete design should be open source and user-friendly to enable non-technical experts to use it (e.g. ecologist).
- The power consumption should be low to enable long operating sessions.
- The monitoring device should be flexible to be combined with various existing traps.
- The insects should be observed from different views and have a uniform background to ease identification afterwards.

3.2 Electrical design

The scheme in figure 3.1 presents the different functions needed for the solution. First, a sensor should detect the presence of an insect and send a trigger signal to the micro-controller. This controller is also responsible for handling the cameras and ask those to take pictures. Finally, after some image processing, the captured pictures should be saved in memory for future observations/analyses. The next subsection addresses the choice of the sensor for the insect detection.

Figure 3.1: Functional scheme of the monitoring device

3.2.1 Possible solutions of sensors to trigger the cameras

To trigger the cameras, we want a sensor capable of detecting efficiently an incoming insect. Hereafter, I present briefly the different possibilities I envisioned.

PIR sensor

It was presented in section 2.1.

Photoelectric beam (Emitter-Receiver)

An emitter and receiver of light (infrared or not) are placed on both sides of the insect path, as shown in figure 3.2(a). When the insect is coming in the trap, it will obstruct the light beam and the receiver will therefore output a difference. To discard environmental perturbations, the light emitted can be modulated [14] (section 7.9.1).
Both emitter and receiver are photoelectric components:

Emitter It is an IR LED. As all diodes, it is made of a PN junction (connection of two semiconductor regions: a p-region, doped with acceptor atoms and a n-region, doped with donor impurities) [26]. Such a structure leads to the diffusion of majority carriers in both directions at the interface. Holes are diffusing from the p-region to the

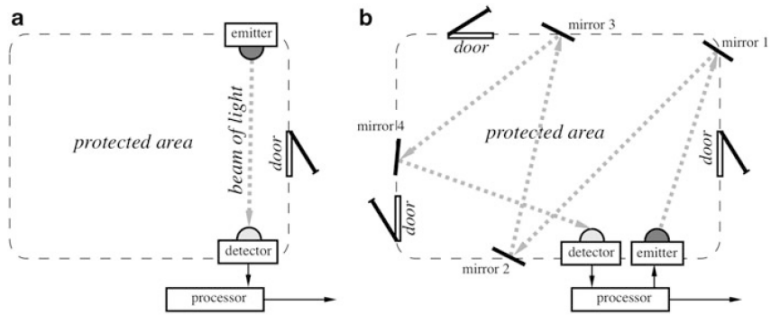


Figure 3.2: Photoelectric-beam interrupters as intrusion detectors. Direct (a) and reflective coupling (b) [14]

n-region and electrons in the opposite way. In the case of LEDs, when we apply a voltage to the junction, we can observe a production of photons. This phenomenon is called electroluminescence. When a voltage is applied, a recombination of an electron-hole pair can occur. If this happens in a direct band-to-band process within a direct bandgap material, photons will be generated.

Receiver The receiver is a phototransistor. This component is based on the inverse principle of electroluminescence. More precisely, a valence electron is elevated to the conduction band after a collision with a photon [26]. This happens when the photon energy (given by $E = h\nu$, with h the Planck's constant and ν the wavelength) is greater than the bandgap energy E_g of the irradiated material (base of the phototransitor in this case). As a consequence, the electron left a hole behind it. The concentration of excess carriers therefore increases under light exposition. Those carriers are now taking part in the current conduction inside the device. This phenomenon is the root of solar panels principle.
Phototransistors are bipolar transistor as shown in figure 3.3. The B-C (Base-Collector) junction is reverse-biased. Under light exposition, electron-hole pairs are generated. Those are swept away from the space charge region and therefore create a photocurrent I_L. The holes that reaches the base p region makes it positive compared to the emitter. The positive polarisation of the junction base-emitter yields traditional transistor effect (electrons are injected from the emitter in the base).

Photoreflector

This is basically the same principle as for the photoelectric beam. In this case, the emitter and receiver are placed side by side. By the use of reflection surfaces such as mirrors, a larger area can be covered by the beam connecting both photoelectric elements (see figure 3.2(b)). Alternatively, the reflective surface can be the intruder itself. The light

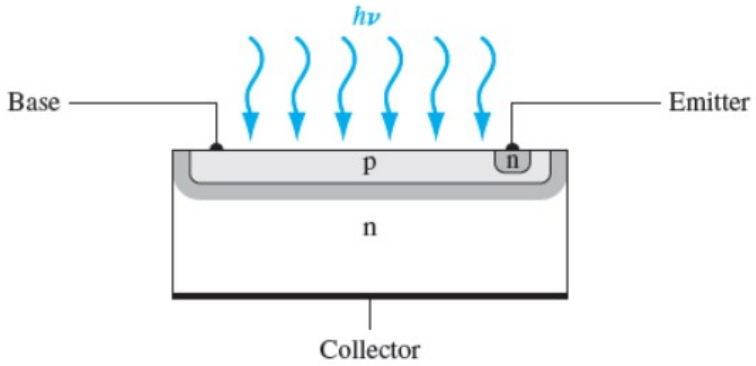


Figure 3.3: Scheme of a bipolar transistor [26]

emitted is reflected on its surface towards the receiver. A detection will be associated to a different quantity of light reflected perceived by the receiver. When there is no intruder, there is no reflective surface in front of the device and the receiver senses none (or few) of the emitted light.

Coupled photodetectors

A photodetector is a device generating voltage as a function of the light it receives (phototransitor e.g.). In opposition to the photoelectric beam and photoreflective sensor just described, this solution would use the ambient light instead of infrared. The principle is illustrated in figure 3.4. The photodetectors sense incoming light from the sun and output a voltage in function of it. The insects are passing in front of only one of those devices (1 on image). When an insect will enter the trap, it will shade the photodetector 1 leading to a different output voltage and thus a detection. As ambient light can vary under environmental conditions (clouds, time of the day, ...), it makes an accurate detection more complicated. The purpose of the second photodetector is to get rid of the environmental perturbations on the sensor signal by differential sensing.

Piezoelectric sensor

A piezoelectric is a material that generates electrical charges as response to a mechanical stress [14] (section 4.6 and 10.4). It is important to note that it works only in Alternating current (AC). So, a transient mechanical deformation will cause an AC electrical response. For our application, we can use a piezoelectric material as a beam-type switch (see figure 3.5), detecting whether or not the beam has been deflected (due to the weight of an object on its tip for example).

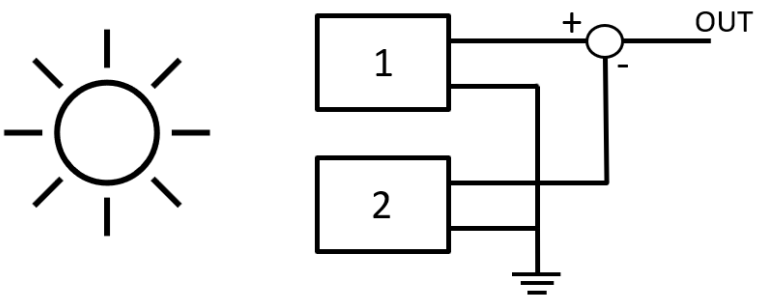


Figure 3.4: Coupled photocell scheme

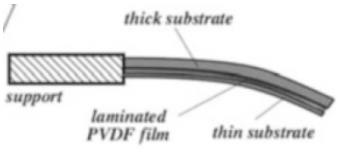
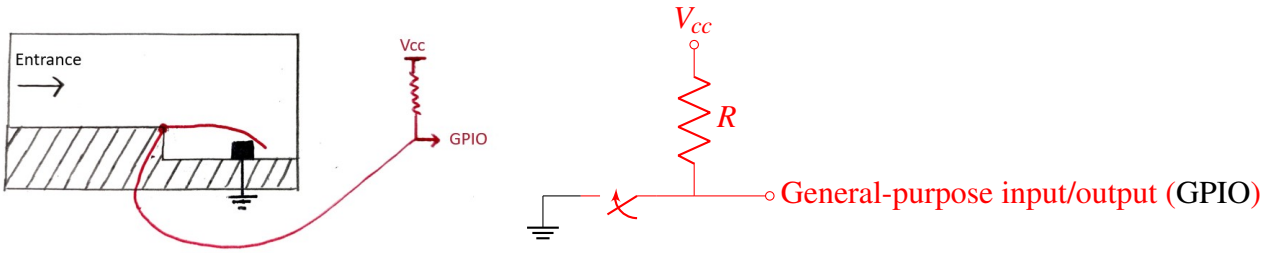


Figure 3.5: Beam-type switch [14]

Contact switch

This is the most simple existing sensor. For this problem, we could use the same layout as for the piezoelectric beam-type switch, putting a flexible electrode instead of the piezoelectric film and another electrode underneath. When the beam is deflected, both electrodes are connected, the switch is closed. The implementation is shown in figure 3.6a. The flexible electrode is drawn in red while the other fixed electrode is in black.



(a) (b)

Figure 3.6: (a) Scheme of the contact switch implementation (b) Equivalent electrical circuit

The interface circuit for a contact switch is really simple : a pull-up/down resistor. For the illustration, figure 3.6b presents the equivalent circuit for the electrodes connected to a pull-up resistor. When the electrodes are not touching, the resistor ensures a non-floating

voltage ($= V_{cc}$) on the output of the circuit (connected to a GPIO pin of the Raspberry Pi). When there is a contact between the electrodes, the output is connected to the ground (black electrode).

Using the camera itself

As the device is using cameras to characterize the insects entering the trap, the triggering could be done by a camera itself. Using an efficient computer vision algorithm to detect the appearance of an insect in the camera field of view and then record some pictures.

Advantages and Drawbacks

To help choosing a solution, some important advantages and drawbacks are gathered in table 3.1 for each sensor.

Sensor	Pros	Cons
1) PIR	+ Widely used for wildlife observation. + Easy implementation. + IR does not perturb wildlife (><visible light)	- Insect thermal radiation should be sufficient to be detected. - Commercial sensors mostly tuned for human detection
2) Photoelectric beam	+ Implementation easy + Long detection range (not needed here) + IR does not perturb wildlife	- Continuous power consumption (though, a LED is not emitting much) - Reduced reliability with fog - Perturbed by sunlight
3) Photoreflector	+ Implementation easy + Emitter and Receiver on the same side + IR does not perturb wildlife	- Continuous power consumption. - Smaller detection range - Perturbed by sunlight
4) Coupled photodetectors	+ Robust to environmental variations : Differential sensing	- Does not work at night - Implementation difficult
5) Piezoelectric beam-type switch	+ Flexible + Implementation easy + Low power consumption Self generating device (no voltage/current supply needed)	- Insect weight should be sufficient to deflect the beam - False trigger due to environmental solicitations (e.g. wind, vibrations) - Works only with transient excitation
6) Simple switch	+ Very simple implementation. + No power consumption between events. + Low cost.	- Insect weight should be sufficient to deflect the electrode - May be fouled by moisture or dust ([14], p. 422). - False trigger due to environmental solicitations (e.g. wind, vibrations)
7) Using the camera itself	+ No additional hardware needed	- Need to process the images online. - Continuous power consumption for cameras. - Complex algorithm.

Table 3.1: Pros and cons of the different sensors

3.2.2 Comparison and selection

Based on the information presented in the previous subsection, it is possible to evaluate the different solutions using a set of criteria. Each solution will be assigned a score

for each criterion: 1 being not suitable at all and 5 corresponds to very good. Table 3.2 contains the score obtained for each criterion and the total score obtained by each sensor. From this comparison, it appears that the Photoreflector is a good choice for the application.

Sensors	PIR	Photoelectric beam	Photoreflector	Coupled Photoreflectors	Piezo	Contact switch	Camera
Power consumption	3	3	3	5	4	5	1
Hardware implementation	3	4	4	1	3	4	5
Complexity	3	4	4	1	3	5	1
Cost	3	4	4	2	3	4	5
Robustness	4	5	5	4	5	2	3
Compatibility μ-Controller	4	4	4	1	4	5	5
Ability to capture all insect types	3	4	4	4	3	3	4
TOTAL	23	28	28	18	25	28	24

Table 3.2: Comparison of the sensors with criteria

As the application does not require large range sensing, it is possible to detect the entrance of an insect using a low power active infrared sensor (reflective).

3.2.3 Interfacing the selected sensor

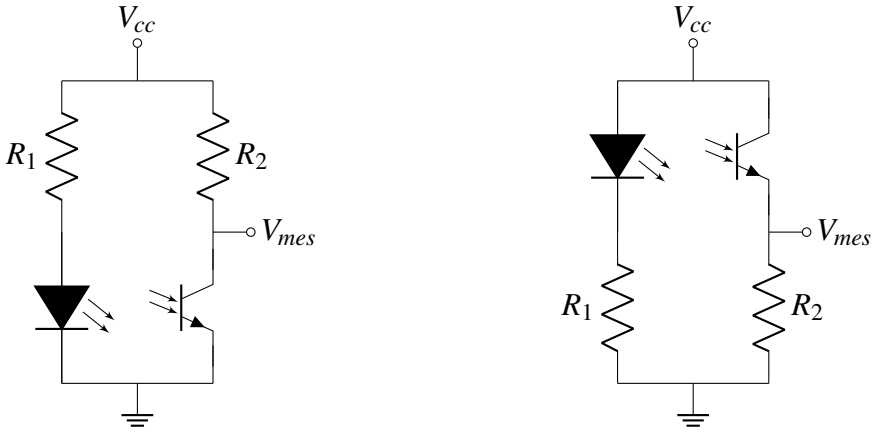
Given the selected sensor, one need to use an interface circuit in order to get a usable reading for the Raspberry PI.

Associating a voltage reading to a high/low reflection

A common way to interface a photoreflective sensor is to use two simple resistors. Those resistors can either be on the high voltage side (as in figure 3.7a) or connected to the ground (figure 3.7b). In the first case, the measured voltage will be near 0 V when the sensor receives lots of the emitted light (turning the transistor ON) and almost equal to the supply voltage V_{cc} when it sees none. In the configuration where the resistors are connected to the ground, a good reflection leads to a voltage near V_{cc} while we measure a low voltage when no reflection can be sensed.
From this information, we have two possible way to detect an intruder in the trap.

1. The sensor sees a low reflectivity surface at rest (black), an intruder will present a higher rate of reflection and trigger the sensor.

2. The sensor is facing a very reflective surface (white). A drop in the amount of reflected light will cause a variation of the voltage measured on the sensor and the intruder is detected.



(a) (b)

Figure 3.7: Interface circuit for a photoreflective sensor (two versions)

Obtaining a binary output

The GPIO pins of the Raspberry can not read analog values. A second stage is thus needed in the interface circuit. A voltage comparator is a circuit that takes two inputs : a variable (in our case, the measured voltage from the sensor) and a reference one. The circuitry is simply composed of an operational amplifier where the inverting input is connected to the voltage measured from the sensor and the non-inverting one to the reference voltage for the comparison. When the variable input (- pin of the Op amp) is higher than the reference (+ pin), the output of the comparator is LOW. It is HIGH in the opposite case. The comparison threshold is a crucial parameter to ensure a good detection rate. The scheme for a comparator circuit is shown in figure 3.8.

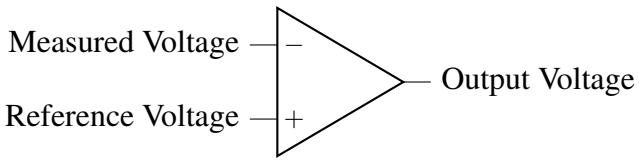


Figure 3.8: Voltage comparator scheme

Breakout board used in the final design

In an effort to provide a complete open source design, it is preferable to use a commercial interface circuit. For line following applications, a robot needs to be able to distinguish a black line from a white surface. This can be done by measuring the reflection of an

emitted IR light on the underneath surface. With this in mind, we can select a commercial breakout board designed for line following applications to apply it to our insect monitoring project. Those breakout boards contains the complete chain from the IR sensor to the digital output compatible with the microcontroller. For design reasons, the product selected was the SEN0017 board from DFROBOT (see appendix A).

The complete electrical circuit of this board is shown in figure 3.9. The two sub-circuits presented earlier can be observed. The interface for the sensor is highlighted in red (1) while the voltage comparator is in green (2). The reference voltage is set by a voltage divider made of two resistor (R_1 and R_5).
In addition, we can find two capacitors connected from the power supply to the ground in parallel (C_1 and C_2). Their purpose is straightforward, capacitors act as a filter letting only AC current go through. Such a capacitor is called a decoupling capacitor. As the DC component is blocked by the capacitor, it will flow through the rest of the circuit while an eventual alternating component will go in the capacitor [27]. From their location, C_1 and C_2 ensure to prevent the small variations that could arise on the supply voltage V_{cc} to propagate to the rest of the circuit. Indeed if this voltage was varying, the sensor output could be varying without being exposed to a different amount of reflected light and possibly cause a false trigger.
There is also a capacitor C_3 connected between the ground and the reference voltage (+ pin of the Op amp). For the same reason as C_1 and C_2, it helps keeping a constant voltage reference for the comparator. Obviously, a varying reference voltage can also lead to false triggers. The LED L_1 (connected through resistor R_4) is just here to inform visually about the output of the circuit.

For line following applications, the threshold of the comparator is fixed to the half of the supply voltage ($V_{ref} = V_{cc}/2$). This corresponds to the highest margin classifier between a black and a white surface.
In the insect detection application, this threshold may need to be different though. At this stage, two options are possible. Either we use the ordered board as such and design the environment of the sensor accordingly. Or we tune the circuit to obtain a different decision threshold. Before deciding which option is the most suitable, it is important to characterize the sensor and the voltage it produces at the entrance of the comparator circuit when some parameters are varying.

Characterization of the selected sensor

The voltage measured at the collector of the phototransistor (pin 4 of the photoreflector U_2 in figure 3.9) is influenced by a lot of parameters : the distance between the sensor and the sensed surface, the reflectivity of this surface (color variation) and the angle of incidence presented by the surface. Hereafter, some tests have been performed to see

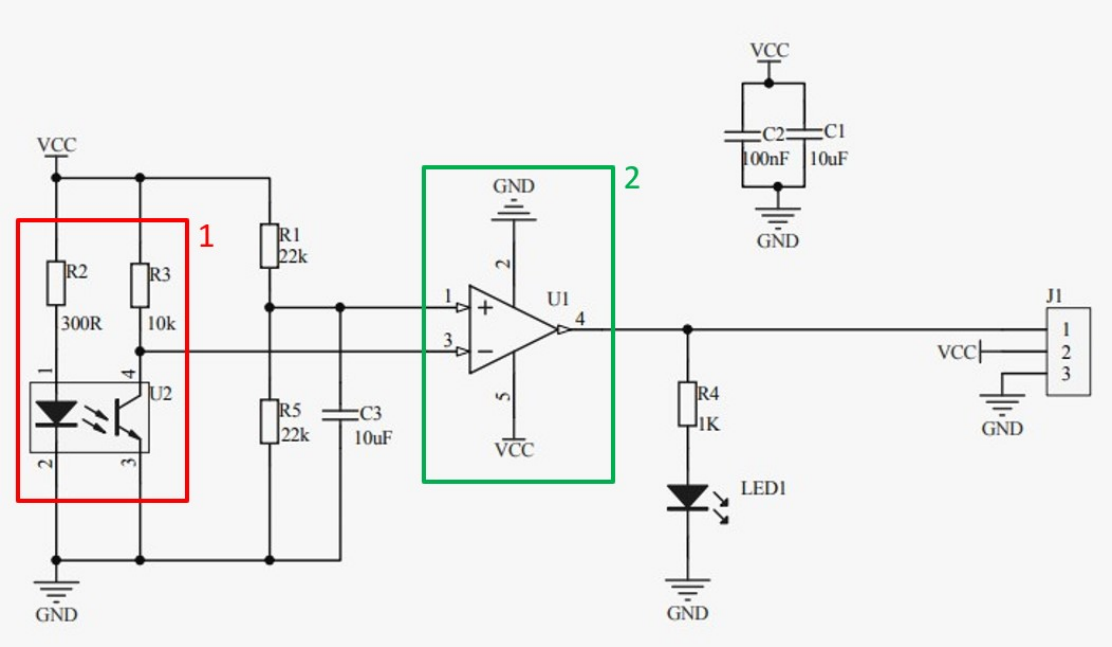


Figure 3.9: Breakout board schematic (see appendix A)

how those are affecting the output voltage.

Output voltage vs distance This first experimental characterization aims to map the voltage measured by the photoreflector as a function of the distance from a flat surface. The color of the surface is obviously influencing this voltage. For this test, the surface used is made of 3D-printing natural acrylonitrile butadiene styrene (ABS) material. The distance is measured using a digital caliper. The voltage is directly measured on the sensor board with a multimeter. The experimental setup is shown in figure 3.10.
The results are plotted in figure 3.11. The sensor is in a configuration such that a low reflection gives a voltage near $V_{cc} = 3.3V$. As expected, the reflection decreases with the distance and the voltage increases.
One can notice that the sensed voltage is higher for really small distances. This phenomenon can be explained by the sensor geometry and its spatial emissivity. From its datasheet [28], the relative radiant intensity is zero for angles higher than 20°. The emitter and receiver are placed side by side at a distance of 2.8 $[mm]$. By trigonometry, it is possible to determine the distance between the sensor and a surface at which the receiver will not perceive emitted IR radiations. More precisely, this minimal distance can be computed as the distance such that a radiation at 20° will reach the receiver:

$$x = \frac{1.4}{\tan(20)} = 3.85\ [mm]$$

This is consistent with the observed voltage increase in Figure 3.11.
The experience was repeated with a surface covered with white paper. As can be seen in figure 3.11, the white paper being more reflective yields higher measured voltages. The low distance effect is even more present with the white surface.

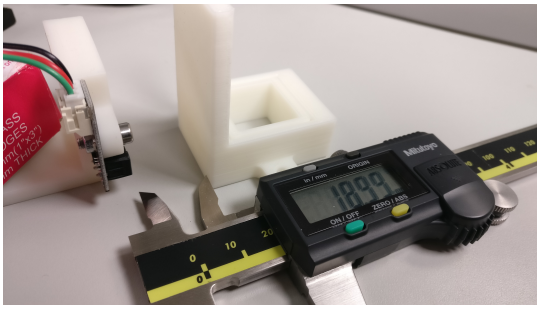


Figure 3.10: Experimental setup : Output voltage VS distance

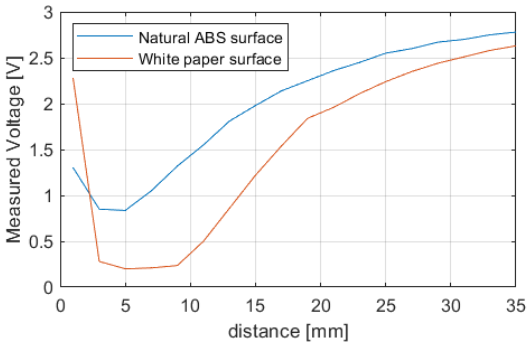


Figure 3.11: Sensor characterization : Output Voltage vs Distance

Output voltage vs gray level Fixing the distance between the sensor and the surface ($\approx 13.5mm$), I placed different color paper pieces to measure the influence of the gray level on the output voltage. The gray levels used for this experiment are expressed in hexadecimal (#FFFFFF corresponding to a perfect white and #000000 to black). As a white surface is more reflective, the measured voltage is lower. It increases as the surface gets darker and saturates at dark levels.

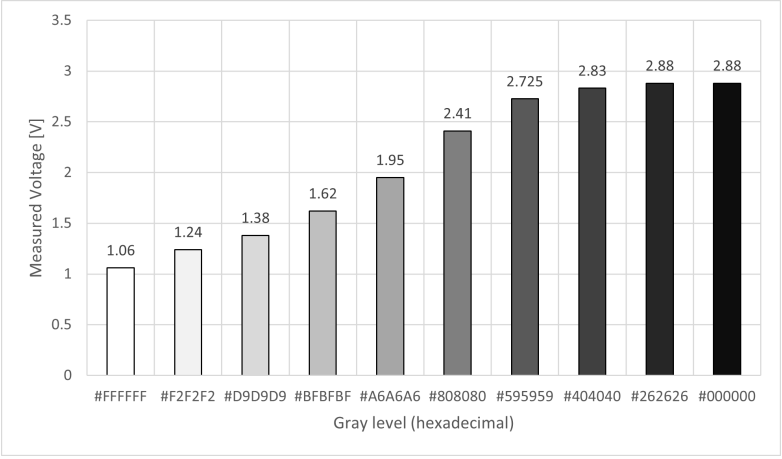


Figure 3.12: Variation of the sensed voltage with the surface gray level

Output voltage vs angle of incidence For this characterization, the goal is to evaluate the impact of the angle presented by the surface in front of the IR sensor. The setup is shown in figure 3.13, the mean distance separating the sensor and the surface is kept constant (15 $[mm]$). The surface is made of white paper. The orientation of the surface is gradually modified by rotating it around a fixed vertical line. The measures were collected every step of 10°, an angle of 0° corresponds to a normal reflection. Results are presented in figure 3.14. It can be observed that the voltage increases with the angle of the surface meaning that the phototransistor receives less infrared radiations.

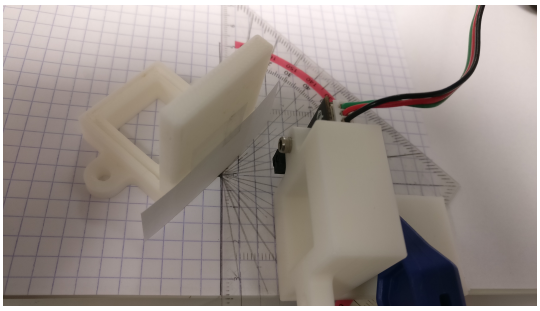


Figure 3.13: Experimental setup : Output voltage VS angle of incidence

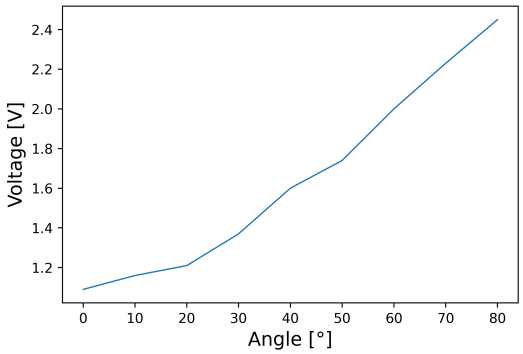


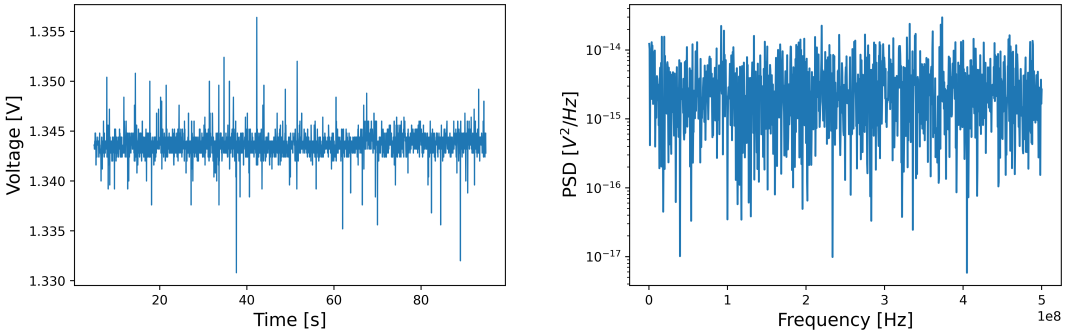
Figure 3.14: Sensor characterization : Output Voltage vs angle of incidence

Noise analysis Measuring the voltage at the collector of the sensor (pin 4 in figure 3.9), it is possible to extract information about the noise. The temporal evolution of this voltage is shown in figure 3.15a. This measurement was realized with an oscilloscope TDS 2004B from Tektronix during a period of 100 [s]. To observe only the noise influence, we keep constant conditions during the experiment: the sensor is placed at the entrance of the trap (as for the final design) at a constant distance of 17 $[mm]$ of a white surface. It can be observed that the voltage is slightly varying around an average value of $1.344[V]$. The noise can be characterized by the standard deviation of the measured signal:

Root mean square (RMS) noise	$1.307[mV]$

This value is negligible compared to the average voltage level ($\approx 0.1\%$). As the voltage threshold for the comparator is at $V_{cc}/2 \approx 1.533[V]$, the noise should not be responsible for false triggers.
The spectral information of the signal can give us different information. The Power Spectral Density (PSD) is computed with the built in python function (scipy.signal.periodogram,

taking the sampling rate of the oscilloscope = $1[GHz]$) and plotted in figure 3.15b. In this case, it is roughly constant all over the spectrum. This corresponds to white noise. The amplitude of this power spectral density is given by $\frac{\sigma^2}{\Delta f}$. If we integrate the mean value of the PSD ($3.415e^{-15}[V^2/Hz]$) over the $5e^8[Hz]$ bandwidth, we obtain $\sigma^2 = 1.7e^{-6}[V^2]$ and $\sigma = 1.307[mV]$ that is the value obtained before with the temporal signal.

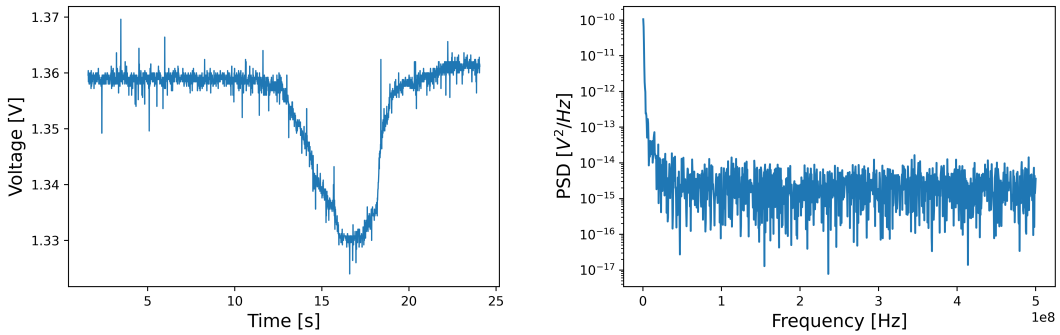


(a) (b)

Figure 3.15: (a) Sensor voltage output over time, (b) Power spectral density of the sensor output voltage

Light influence The following experiment tends to evaluate the impact that ambient light has on the sensed voltage. The experimental setup is the same as for the noise analysis presented in the previous paragraph, an external lighting source (phone torch light) is swiping gradually the entrance of the trap. Figure 3.16a shows the voltage measured at the collector of the phototransistor when gradually increasing the illumination around the sensor. The minimum voltage (≈ 1.33 $[V]$) corresponds to the moment where the light was perfectly aligned with the entrance of the trap, hence best illuminating the sensor. As in the previous experiment, the associated power spectral density is computed(figure 3.16b). The resulting graph is similar to the one with constant illumination (figure 3.15b) except in the low part of the spectrum. We can indeed observe a higher PSD for lower frequencies. This is typical for environmental perturbations : slow variations as shown in figure 3.16a. The insect monitoring device could be subject to such variations when installed during several hours in the nature. The sensor is located near the entrance of the trap and is then subject to illumination changes (sunrise, sunset ...).
The maximum deviation observed in this test is of ≈ 30 $[mV]$ and should not be responsible for false triggers. A solution to discard environmental perturbation if those were threatening the sensing performances is to use a second identical sensor. The latter is subjected to the same light irradiation but would be placed out of the insect passage way thereby sensing only environmental perturbations. By subtracting one signal from

the other we can discard the environmental perturbations. This is another example of differential sensing.



(a) (b)

Figure 3.16: (a) Sensor voltage output over time under varying ambient illumination, (b) Power spectral density of the sensor output voltage under varying ambient illumination

Detection with the selected sensor

Based on the characterization of the sensor, the sensor is putted at the entrance of the imaging tunnel at a distance of ≈ 17 $[mm]$ from a white paper surface. The voltage measured on the sensor will thereby lie just below the comparator threshold ($V_{cc}/2$). Once an intruder will cross the sensed field, the voltage will exceed this thresold and result in a detection. The waveforms associated to a detection of an object (finger) inside the trap entrance are shown in figure 3.17. This was measured with an oscilloscope TDS 2004B from Tektronix.

3.2.4 Camera

The cameras used for this project are Universal Serial Bus (USB) Digital Microscopes from Plugable (reference in Appendix A). Those offers a manual focus setting, a built-in tunable lighting and a high-definition of up to 2 MPixels. They are low cost and their USB connection makes them easy to interface with microcontrollers.

3.2.5 Microcontroller

The raspberry Pi is a low cost, widely commercialized option. I use a model 3B+ for this project. Compared to an Arduino, it offers several USB port needed to connect the USB cameras. The numerous GPIO connectors enable interfaces with the other electrical

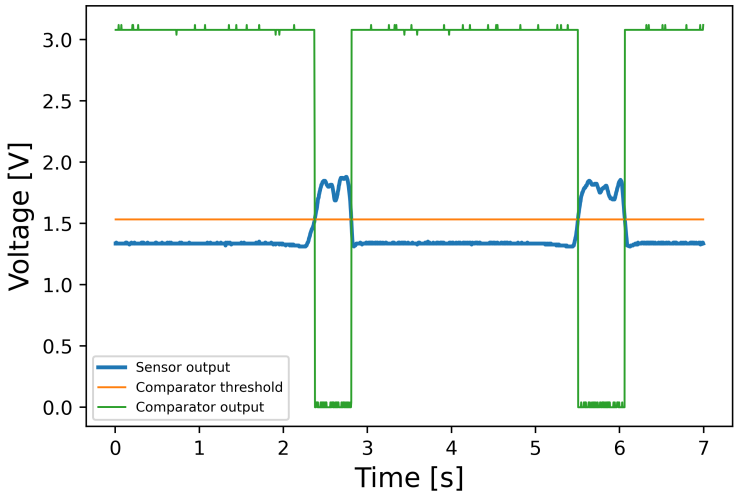


Figure 3.17: Waveforms associated to a detection

components (sensor board and LEDs). Figure 3.18 illustrates the connections between the raspberry Pi and the other components. When needed, GPIO can be configured as output (generating HIGH or LOW voltage for the LEDs or powering the sensor) or as input and read from the sensor.

3.3 Structural design

This section gathers information about the structure of the monitoring device. First, the imaging box structure is presented in subsection 3.3.1 and then subsection 3.3.2 explains how this can be integrated with an existing insect trap.

3.3.1 Imaging box structure

This section is about the design of the structure that will hold the different cameras, the detection sensor and different elements needed for the observation of insects. The different specifications needed for this design are gathered hereafter:

1. The insect pictures should be taken from different views
2. Each picture should be taken with a plain (white) background
3. Each camera should not interfere with the others (clear vision field)

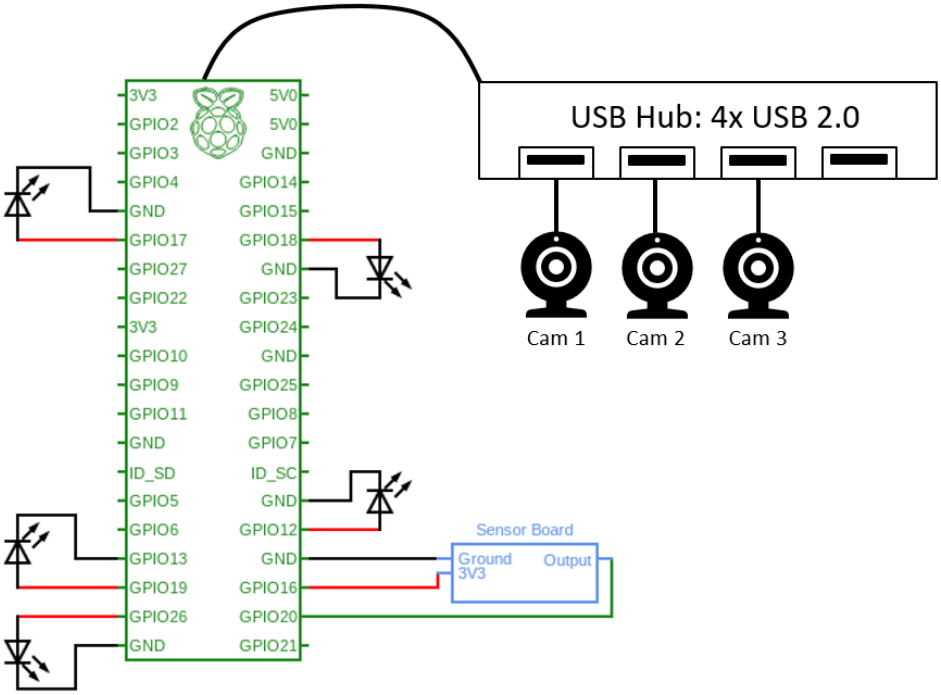


Figure 3.18: General electric scheme

To comply with condition 3 and to evaluate the size of the background elements, it is necessary to model the field of view of the cameras. To this end, some experimental characterization were performed. The size of the projected field of view on a sheet of paper at different distances from the camera was calculated. The measured distances and sizes are reported in figure 3.19. The circle represents the edge of the camera (from where was measured the distances to the paper sheet). From this reference, three measurements of the projection where made (the rectangles on the picture).
One can notice that the field is rectangular and not squared so to describe the field of view, we need two angles. Those can be obtained by trigonometry from the measurements. Those could have been obtained with only two data points but it was decided to work with 3 and take an average of two angles computed to be more rigorous. The angle describing the increasing of the window size with the distance along the horizontal plane is $\alpha_1 = 5.85°$ and the one in the vertical plane is $\alpha_2 = 4.55°$. As the length of the projection window is greater along the horizontal plane than along the vertical one, it is consistent to have a greater angle α_1.
For the rest of the design, a truncated rectangle-based pyramid (designed in Solidworks) will be used to visualize the field of view of the different cameras.

Three angles of view are used for capturing insect images: one below, one top and one

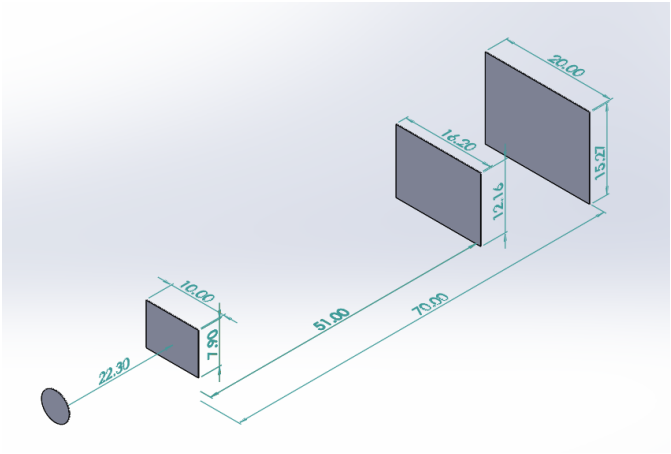


Figure 3.19: Measurement to characterize the camera field of view

side view. The principle of the device is the following: the insect is captured by an existing trap where it is brought to a glass tunnel in our device. The connection between the existing trap and our structure is detailed in next section. The glass tunnel is made of microscope slides mounted together to form a squared pathway (see figure 3.20). The IR sensor covers the entrance of the field as highlighted by the red arrows. The three cameras are then articulated around the tunnel to capture the insect from different views. The complete structure is illustrated in Figure 3.21.

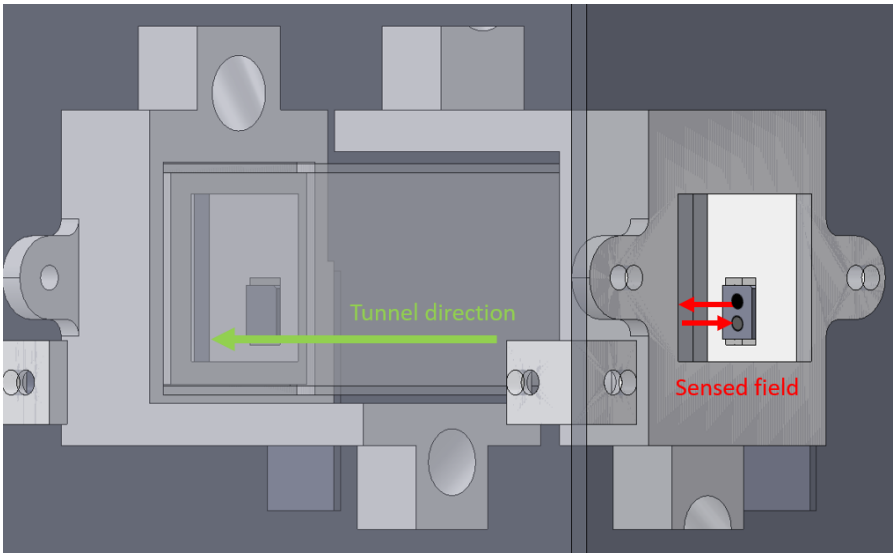


Figure 3.20: Glass tunnel and entrance sensor

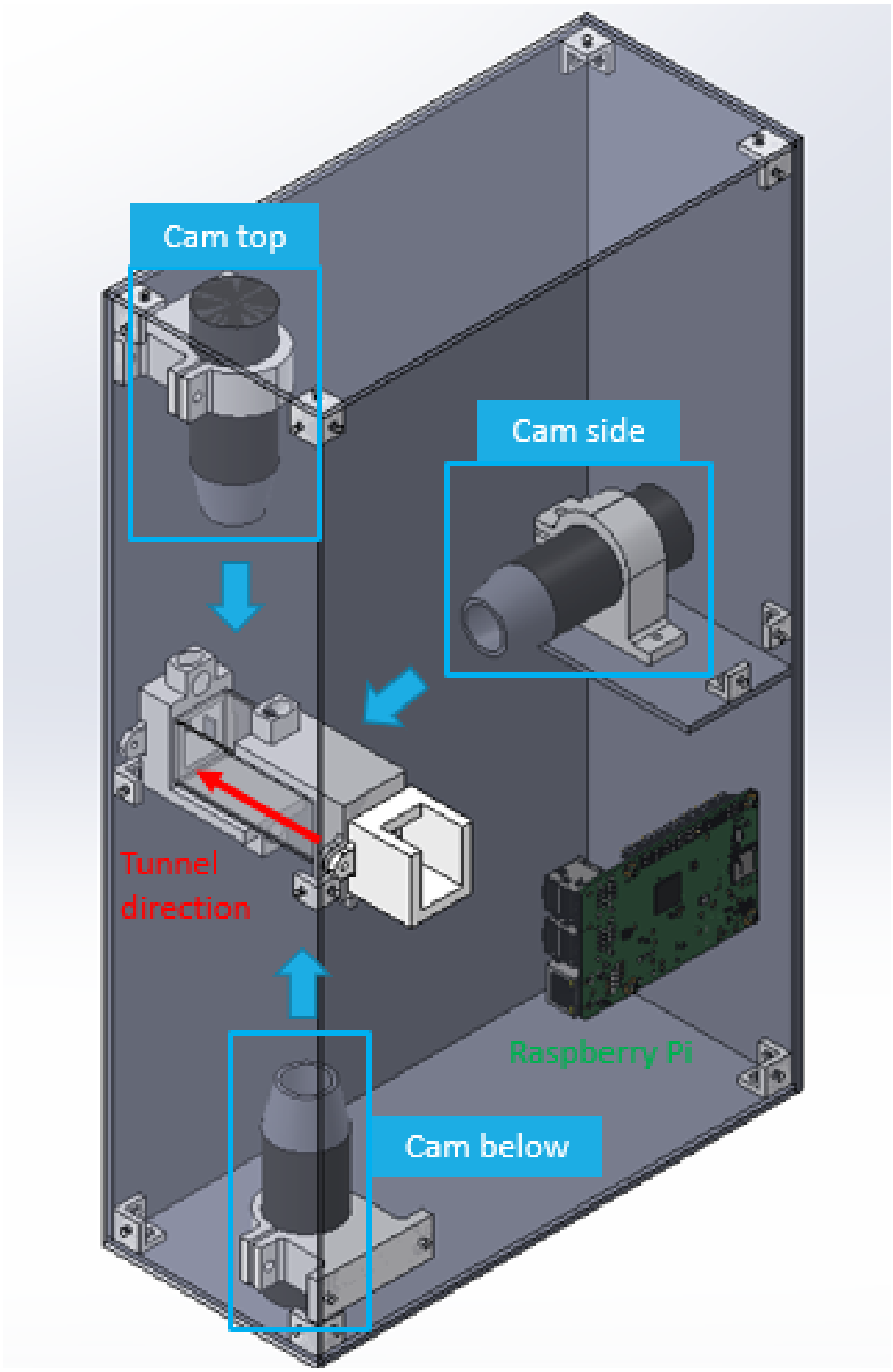


Figure 3.21: Isometric view of the final design in Solidworks

The structure is made up of acrylic plates that were laser cut. 4 different plates can be distinguished in the design (see Figure 3.22). Another plate is used to support the side view camera as shown in Figure 3.21.
The technical drawings of those plates are gathered in Appendix B. To assemble those plates, two different kind of support pieces are used : L-support and angle support (Figure 3.23a and Figure 3.23b). Those parts were 3D-printed, they present hexagonal holes to hold nuts.
The camera holders were also 3D-printed (Figure 3.23c). The holders for the three cameras are not exactly similar because their height is adapted in function of their position relative to the glass tunnel. Finally, the entrance and exit parts of the tunnel are printed as well (Figure 3.23d). They are designed to hold the 4 glass slides and the IR sensor. A plane surface extends the part to obtain a white background for the pictures. Holes are designed to hold LED's for the lighting.

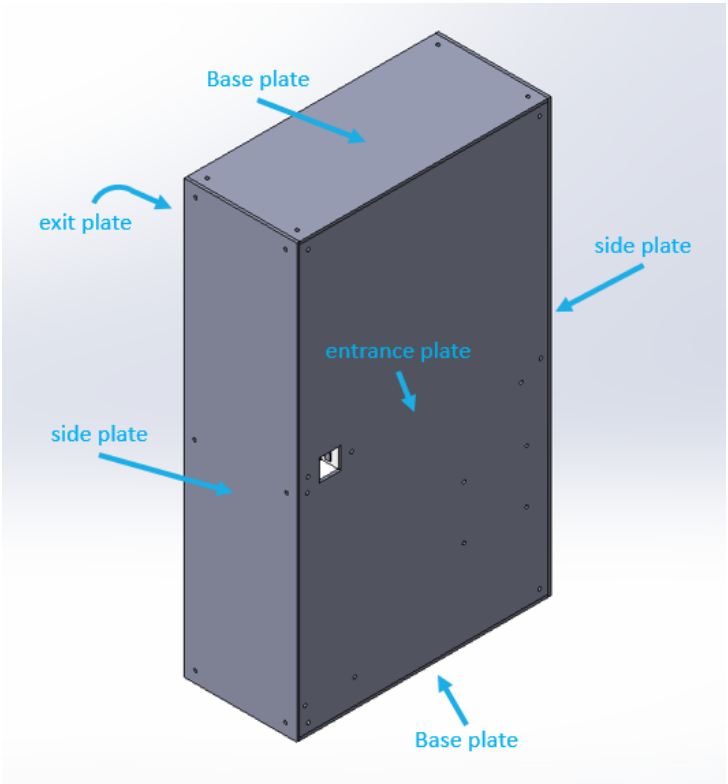
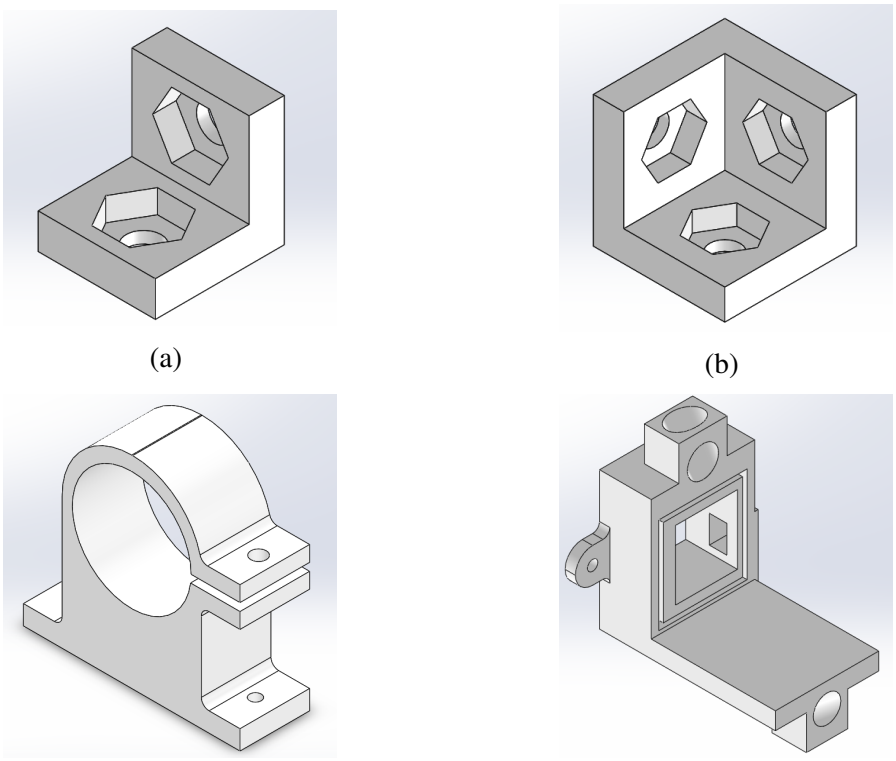


Figure 3.22: Different acrylic plates of the structure



(a) (b)

(c) (d)

Figure 3.23: (a) L-support, (b) angle-support, (c) Camera holder, (d) Entrance part

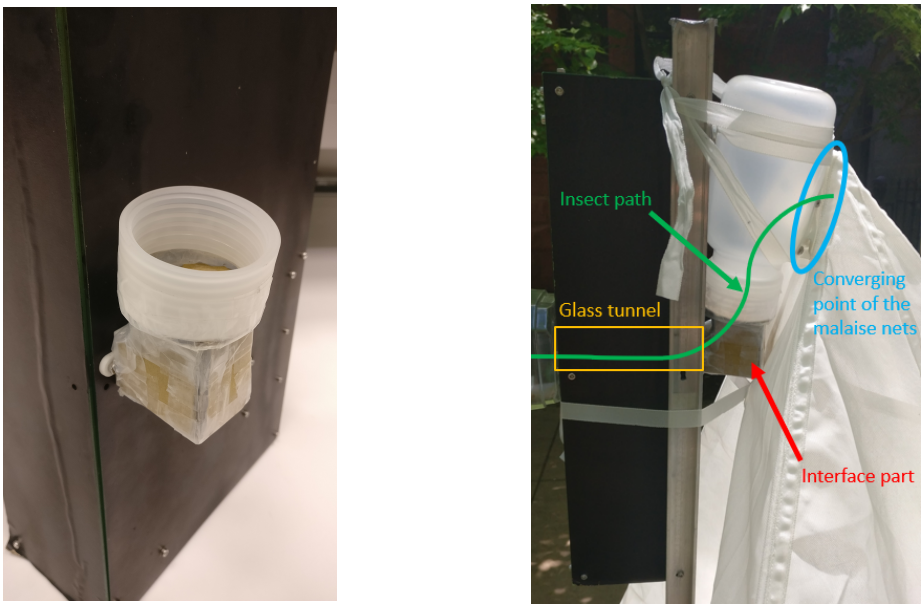
3.3.2 Trap integration

For this thesis, we decided to test the structure on a Malaise trap as presented in section 2.2. A commercial Townes-type malaise trap (more details in Appendix A) was chosen. A more affordable alternative is to use a "home-made" malaise trap as presented in [29]. The traditional use of a malaise trap is to connect the output of the nets to a collecting bottle filled with ethanol. This solution kills the insect while preserving their DNA for later analyses. To integrate the monitoring structure presented in subsection 3.3.1, we need to connect the output of the malaise trap nets to the box entrance. This connection is done via an interface piece as illustrated in Figure 3.24.
To set the trap up, we need the material showed in Figure 3.25:

- The malaise trap nets
- The monitoring device
- Two poles (180 cm and 120 cm)
- 14 stakes

- Rope (to attach to the stakes to fix the nets)
- The interface piece

The monitoring device is screwed to the larger pole, the top of this pole is pierced as depicted in Figure 3.26a to fit with the holes of the side plate. The connection of the monitoring device to the pole is shown in Figure 3.26b. Finally, Figure 3.27 summarizes the integration of the monitoring device with the malaise trap by showing the complete setup. Tables 3.3 and 3.4 contain all the laser cut and 3D-printed parts.



(a) (b)

Figure 3.24: (a) Interface piece, (b) Connection of the malaise trap to the monitoring device

Part	Amount
Bottom plate	2
Side plate	2
entrance plate	1
Exit plate	1
Transverse support	1

Table 3.3: Laser cut parts

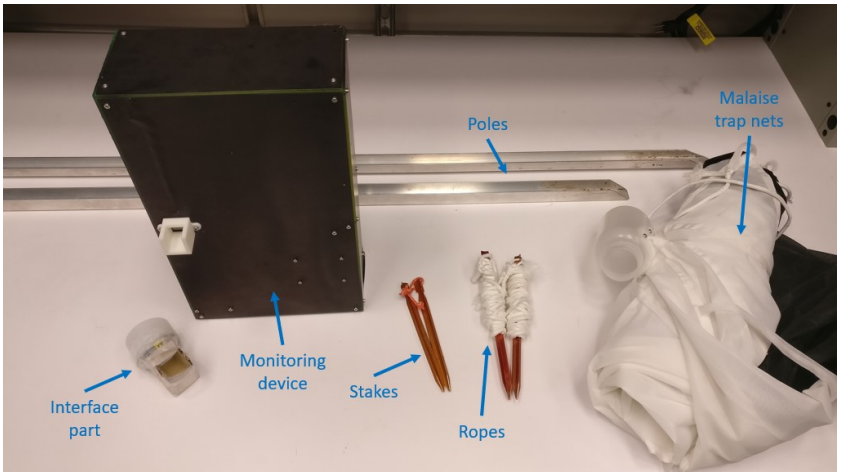
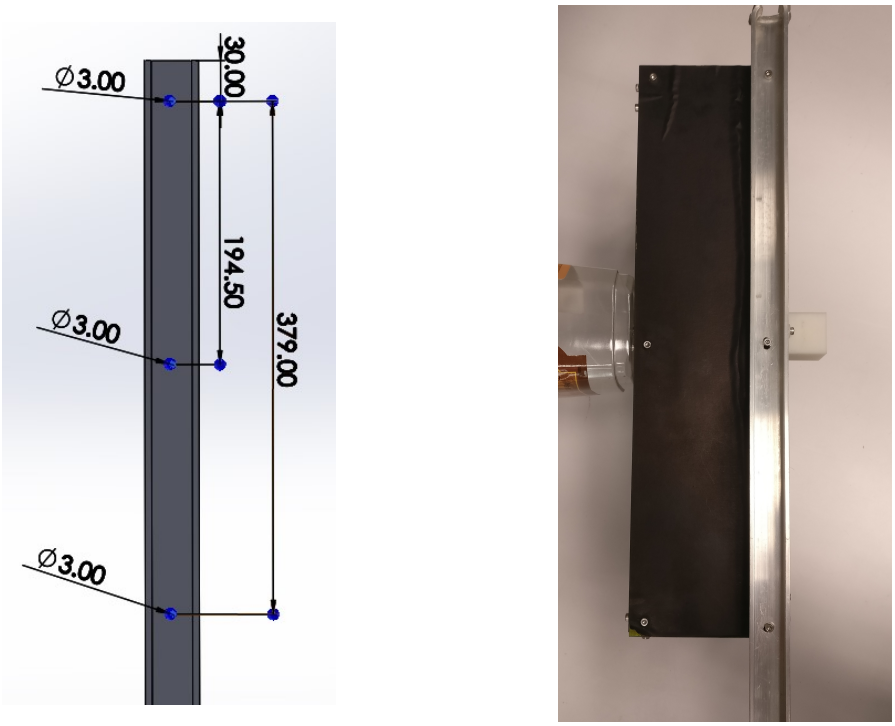


Figure 3.25: List of material to set up the trap



(a) (b)

Figure 3.26: (a) 180 cm-pole piercing pattern, (b) Connection of the monitoring device on the pole

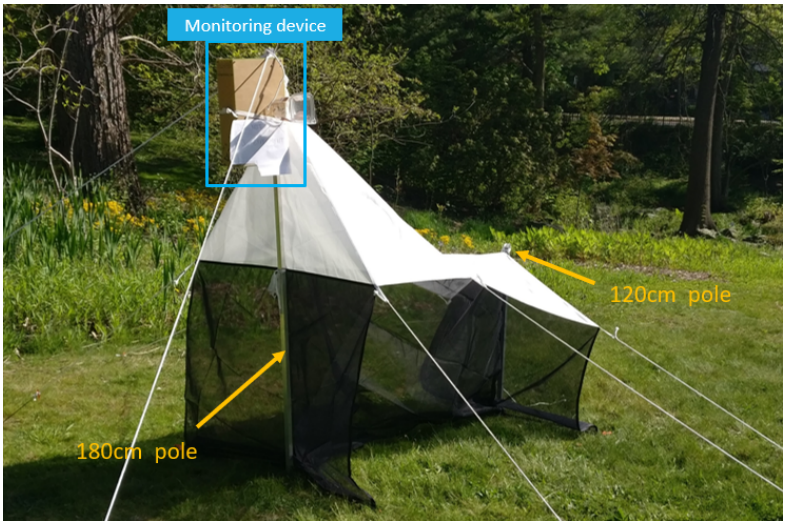


Figure 3.27: Complete trap integration

Part	Amount	mass per unit [g]	Total mass [g]
Camera holder	3	19.42	58.26
Angle support	8	1.03	8.24
L support	7	0.46	3.22
Entrance part	2	20.08	40.16
Total			**109.88**

Table 3.4: 3D-printed parts

3.4 Cost of the device

To build the insect monitoring device, it is necessary to buy, print and laser cut different components. Table 3.5 summarizes the costs associated to each purchased part (more information can be found in Appendix A). The cost to laser cut the different plates should not exceed 10$. Finally, we can compute the cost to print the pieces using their mass. 1 spool of ABS material for 3D printer of 0.65 $[kg]$ costs 69 [$] (Buy - ABS for 3D printers), giving a cost per gram of material of 0.106 $[\$/g]$. The total mass of ABS material used was computed in 3.4, this corresponds to a cost of 11.66 [$]. To print the different pieces, support material was often required. It can reasonably assumed that for the printing of the different pieces, the mass of support material required was half of the one of the actual piece. 54.94 $[g]$ of support material is thus necessary. The mass cost is 0.264 $[\$/g]$ (Buy - Support material for 3D printers). The cost for support material is

then 14.53 [$]. Finally, the total cost for the construction of the monitoring device is:

$$\begin{aligned} C_{tot} &= C_{purchase} + C_{lasercut} + C_{print} \\ &= 443.22 + 10 + 11.66 + 14.53 = 479.41 \text{ [\$]} \end{aligned}$$

Component	Amount	Cost per unit [$]	Total cost [$]
Raspberry Pi 3 model B+	1	35	35
Plugable USB 2.0 Digital Microscope	3	33.95	101.85
SEN0017 Sensor	1	5.9	5.9
Battery	1	24.99	24.99
Malaise trap	1	196	196
Acrylic plate	3	13.69	41.07
LED	5	6.95 (pack of 25)	6.95
Screws			
M3 L6	39	8.86 (pack of 100)	8.86
M3 L12	13	10.37 (pack of 100)	10.37
M2.5 L8	4	6 (pack of 100)	6
Nuts			
M3	52	2.33 (pack of 100)	2.33
M2.5	4	3.9 (pack of 100)	3.9
Total			**443.22**

Table 3.5: Purchased components: costs

Chapter 4

Software

This chapter focuses on the software implementation of the desired behavior for the monitoring device presented in chapter 3. First, an overview of the code structure is presented in section 4.1. The following sections describe the execution sub-blocks step by step . Section 4.2 explains how to read the detection sensor information and what happens when an intrusion occurs. Then, the image acquisition process is described in section 4.3. Section 4.4 eventually deals with the post-processing performed before saving images in the microcontroller memory.

4.1 Overall structure

The overall structure of the code is organized in threads to enable concurrent execution of different part of it. More precisely, there are two main threads:

1. **"Sensor thread"** : In this thread, we associate a callback function to the interrupt signal of the IR sensor. At the moment the sensor detects an intrusion, this function is called.

2. **"Camera thread"** : waits for a signal coming from the other thread to start image acquisition.

The principle is illustrated in Figure 4.1. The handling of the cameras and the IR sensor is splitted in two communicating threads. Once the detection happens, a trigger signal is passed to the other thread.

4.2 Detecting an insect with the IR sensor

As first step in the code of the monitoring device, we need to deal with the signal coming from the sensor board. The output of the SEN0017 sensor board (appendix A) is high

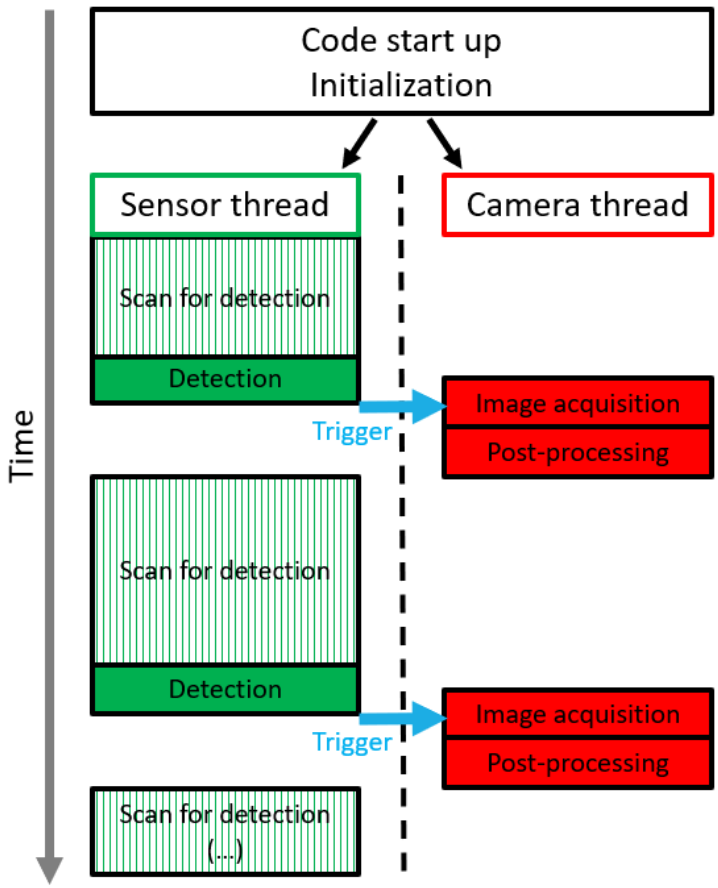


Figure 4.1: Code Architecture

when there is nothing in front of the sensor except the white reflective surface. When an insect enters the trap, the voltage drops because the threshold is crossed. A falling edge on the GPIO pin connected to the sensor board is thus associated with the detection of an insect. Algorithm 1 associates this interrupt signal to a callback function that is called each time this event occurs.

Algorithm 1: Sensor Handling

Assign: Insect_detected_callback ← Falling edge event on GPIO 20;
As soon as a Falling edge occurs on GPIO 20,
Insect_detected_callback is called
Function `Insect_detected_callback():`
  Lights → ON;
  Send {Trigger} → Camera_thread
End Function

4.3 Image acquisition

Inside the thread handling the cameras, as soon as the sensor interrupt callback function is triggered, each camera captures a bunch of pictures (30 each) following the detection. This section presents first how the cameras are interfaced with the microcontroller. Then, different possibilities are investigated and compared to capture the 30 pictures. Finally, we will see which settings are important for the camera devices and how it affects the resulting images.

4.3.1 How are the camera peripherals interfaced?

The cameras are USB microscopes, they are connected to the USB hub of the Raspberry Pi. This USB hub is communicating with the CPU via a USB bus as can be seen in figure 4.2. The USB bus has a limited bitrate of 480 $MBit/s$ or 60 MB/s that will limit the image acquisition as well [30]. Considering that the resolution of the images is 640x480, that the pixel values are encoded as a byte (value between 0 and 255) and that we have 3 channels for the colors, the size of one image is therefore :

$$size_{image} = 640 * 480 * 3 * 8 = 7.3728MBit = 921.6kB \tag{4.1}$$

In practice, the USB bus can never transfer data at its maximal rate. A maximum rate of 30 MB/s is a more reasonable estimation. It is even less when the transfer includes lots of small files (as it is the case in this application). Based on this information, we can determine the maximum framerate at which we will be able to capture images with 3 simultaneously working cameras:

$$fps_{max} = \frac{30\ MB/s}{size_{image} * 3} = 10.85fps \tag{4.2}$$

To handle video streams, the Raspberry Pi is using Video for Linux 2 (V4L2) which is the Linux kernel driver for images/videos related devices [31],[32]. It ensures the mechanism to pass video frames from one device to another. For example, capturing frames from a webcam device to save them in memory.

4.3.2 Different methods to acquire images

FSWebcam

At this stage, the code should be able to fetch some pictures from a specific device (one of the three cameras) and save them in the wanted file. The library FSWebcam provides a bunch of functions to perform the desired actions (see more information in Appendix C). The command to save an image can either be blocking or not. When it is not, it runs in

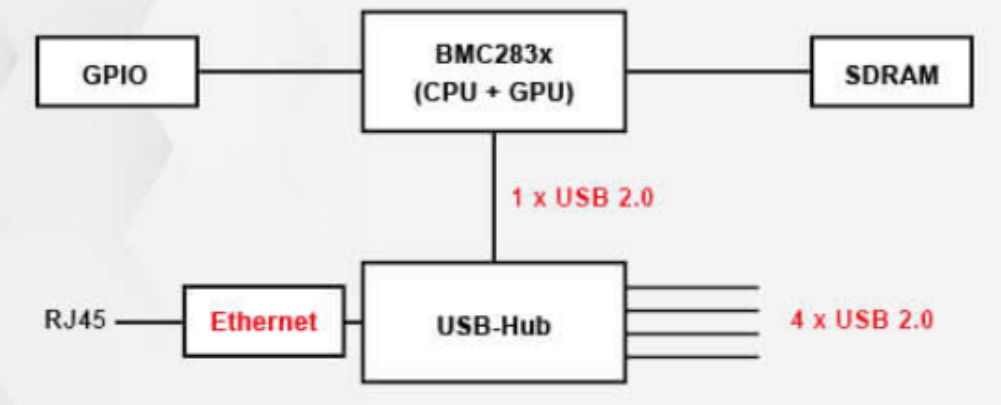


Figure 4.2: Raspberry Pi USB [30]

background and the code continues to execute afterwards. Running in background can be useful to parallelize the execution on the different cameras. Nonetheless, if the code tries to save a new image from a busy device, the previous operation will be aborted. It is therefore important to ensure that the saving process has been completed.
As benchmark to evaluate the time performance of the image acquisition, we will measure the time that the algorithm takes to save 10 images from each camera once the sensor has been triggered.

- Repeating 10 times the acquisition of images from the three angles within a loop takes roughly 9.5 [s]. At each iteration, the first two cameras are called to run in background and the last one is called with a blocking statement to ensure the completion of the saving process. This is illustrated in Figure 4.3 and the corresponding pseudo-code is shown in algorithm 2.

Algorithm 2: Image Acquisition FSWebcam v1

Trigger : Event on Sensor GPIO
for $N = 1 \rightarrow 10$ **do**
  Take picture {camera 1, background}
  Take picture {camera 2, background}
  Take picture {camera 3}
end for

- To speed up the execution, each camera can be running in a different thread. If it is the case, each image acquisition command should be blocking (otherwise the camera will start taking second picture before saving the first one). With this implementation, the time taken to save 10 pictures from each view is lowered to $\approx 8[s]$. The code within one thread is shown in algorithm 3. Figure 4.4 illustrates this.

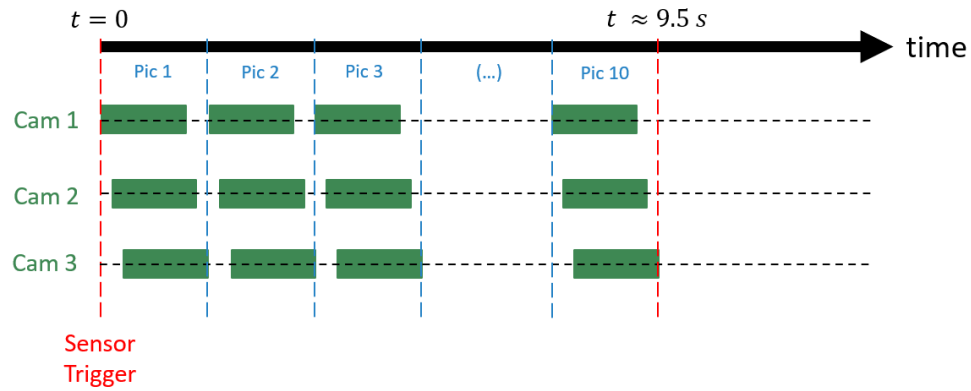


Figure 4.3: Timing of image acquisition using Algorithm 2: Image Acquisition FSWebcam v1

Algorithm 3: Image Acquisition FSWebcam v2

```
Same code for each thread
Trigger : Event on Sensor GPIO
for N = 1 → 10 do
    Take picture {camera}
end for
```

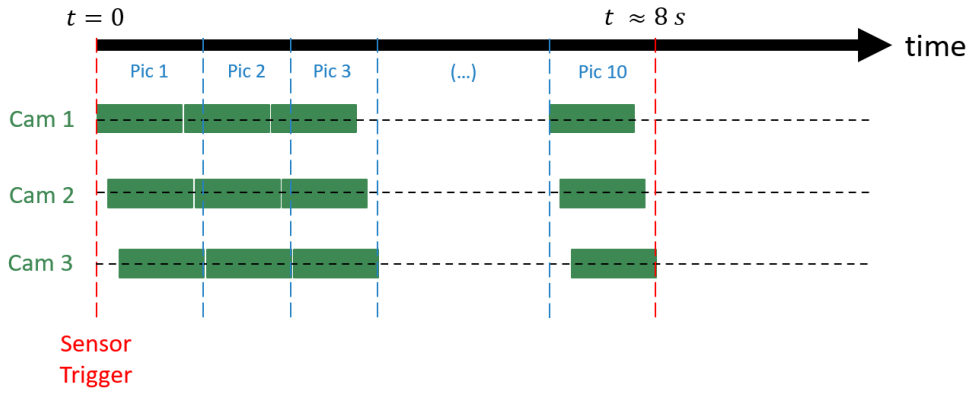


Figure 4.4: Timing of image acquisition using Algorithm 3: Image acquisition FSWebcam v2

FFMPEG

As it turned out that taking pictures with FSWEBCAM was slow, it is interesting to look for an alternative. With FFMPEG (see more information in Appendix C), we can ask the Pi to extract frames from the video flux of the cameras and save them as images in memory. This is described in pseudo code in algorithm 4.

Algorithm 4: Image Acquisition FFMPEG

Trigger : Event on Sensor GPIO
In each thread:
Save ← Extract frames {video flux [cam_ID]}

This version of the image acquisition code enables to reach a much better performance in terms of time and rate: the concurrent acquisition of 30 pictures on each camera takes $\approx 5.2\ [s]$. When the different threads run at the same time and ask the camera to save picture simultaneously, the data flow is overwhelmed and the driver keeps asking the camera to retake the frame. This can lead to very long execution time and the insects passage can be missed. To counter this problem, a solution is to introduce a temporal offset between the cameras. Choosing an adequate value for this time delays cancels the previously mentioned phenomenon. The timing of this image acquisition method is illustrated in a diagram in figure 4.5. The cameras 2 and 3 are activated with a delay of $4/7\ [s]$ and $8/7\ [s]$ respectively after the IR sensor trigger. Those delays were chosen by trial/error to ensure a rapid execution that does not overload the USB bus. This method enables to capture 3 times more pictures in half the time compared to the methods using FSwebcam and is thus selected as image acquisition method for the project.

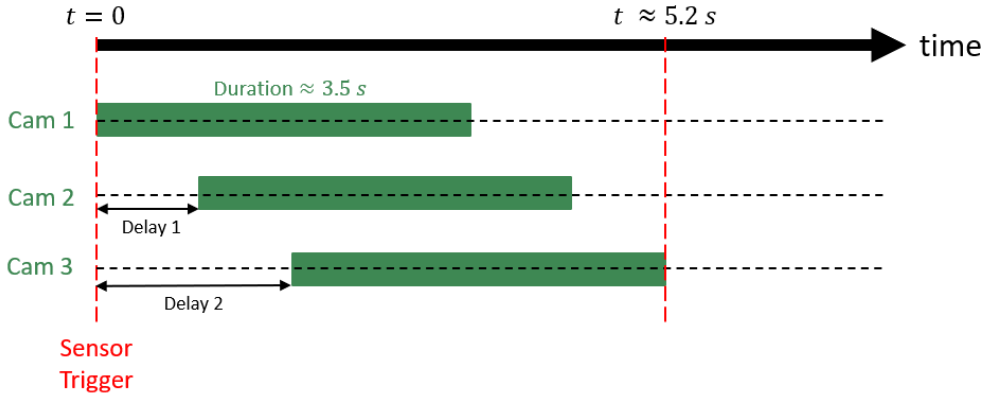


Figure 4.5: Timing of image acquisition using algorithm 4: Image Acquisition FFMPEG

4.3.3 Camera settings tuning

The image acquisition process also implies setting parameter values to obtain good quality images. The parameters can be set using V4L2. In this project, the parameters of interest were:

- `exposure_auto (menu)`: sets the exposure mode to Manual (1) or Aperture priority mode (3). In our application, a manual exposure is preferable to obtain constant conditions from image to image.

- `exposure_absolute (int)`: sets the exposure time (1 = 100 μs) if the exposure mode is Manual. This parameter is crucial, a longer exposure will give brighter image but will also increase the blur in case of motion in the field of view. This parameter needs to be set by finding a trade-off between brightness and sharpness. A solution to decrease exposure time while keeping sufficient brightness for insect observation is to bring more light during image acquisition.

- `gain (int)`: This setting controls the gain of the camera sensor (signal amplification). This should be handled cautiously because a higher gain leads to brighter images but also an amplified noise [33].

- `gamma (int)`: The gamma parameter is characterizing the camera encoding non-linearity. The eye perception of image intensity is non-linear. A simple linear encoding of the pixel intensity is therefore not adequate to fit the eye perception. This can be observed in Figure 4.6. A gamma correction ensures a more efficient tone distribution taking the eye perception into account [34]. Such a transformation takes the following form:

  $$V_{out} = V_{in}^{\gamma} \tag{4.3}$$

  If $\gamma > 1$, a broader range of values is assigned for brighter pixels. When $\gamma < 1$ on the other hand, darker pixels are prioritized in the tone distribution. A linear encoding camera ($\gamma = 1$) maps the number of photons sensed into output signal proportionally.

Besides, other parameters can be changed: `brightness`, `contrast`, `saturation`, `hue`, `white_balance_temperature`, `sharpness` and `backlight_compensation`. In this project, these settings are fixed to their default value. The following command is used to modify a parameter value:

```
v4l2-ctl -d <device> -c <parameter>=<value>
```

For example, to set the exposure to manual with an exposure time of 15 $[ms]$ for the first camera (accessed as `/dev/video0`), the command would be :

```
v4l2-ctl -d /dev/video0 -c exposure_auto=1 -c exposure_absolute=150
```

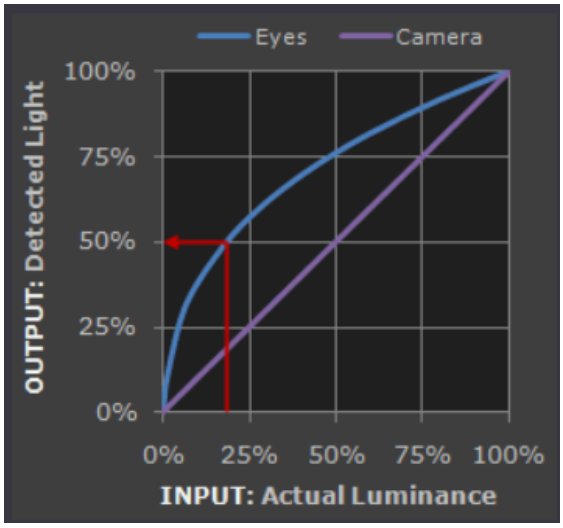


Figure 4.6: Light perception for eyes and linear encoding camera [34]

4.4 Image post-processing

4.4.1 Selecting an image

Once an insect has entered the trap and triggered the entrance sensor, the three cameras are capturing a set of images as explained in section 4.3. This method is simple but implies the record of 90 pictures per insect visit (30 per camera). These pictures contains moreover very likely useless frames (partially visible insect e.g.). To save storage space, it would be wise to keep only one frame per camera and per insect visit. This selected picture should maximize the insect visibility. Figure 4.7 illustrates how this can be applied to the insect monitoring application. Once the set of images related to one insect visit is taken, the algorithm compares each of them with a reference image using the maximum visibility criterion. The reference image is taken when no insect is inside the tunnel. The selected image is finally saved in memory while the others are deleted.

To implement this maximum visibility criterion, different algorithmic possibilities were investigated. Their basic principles are presented hereafter. For the 4 different methods, the images are compared in gray scale. The images can directly be read in gray scale using OpenCV library (Appendix C).

Mean Squared Error (MSE) criterion

This is the most famous error criterion and it is used in many applications (loss function for regression models e.g.). In image applications, we can compute the difference between two images using the MSE definition [35]. Let us denote the two input images

Algorithm 5: Most relevant image selection

```
Input : {Set of captured images S}, IM_REF
score_ref ← 1
index ← 0
for N = 1 → IM_COUNT do
    score ← EVALUATE_CRITERION { S[N], IM_REF}
    if score > score_ref then
        score_ref ← score
        index ← N
    end if
end for
Save {S[index]}
```

x and y. Considering each image as an array of m by n pixels, the MSE is computed as

$$MSE(x,y) = \frac{1}{m}\frac{1}{n}\sum_{i=1}^{m}\sum_{j=1}^{n}(x_{ij}^2 - y_{ij}^2) \tag{4.4}$$

The maximum visibility criterion is obtained as the maximum MSE computed between the taken pictures and the reference.

Structural Similarity Index (SSIM)

Another way to define the maximum visibility criterion is to use the SSIM presented in [36]. This method is used for example as image quality assessment comparing a distorted to a reference picture. The index is a score characterizing the similarity between the two image inputs. A score of 1 corresponds to a perfect match while -1 is the worst score possible.
Unlike using a MSE metric, the Structural Similarity Index proved to better fit the Human Visual system (HVS) for image quality assessment. Previous attempts to mitigate the limitations of MSE criterion were based on the quantification of error visibility. Those implementations try to simulate the properties of the HVS which were established through psychophysical and physiological experiences. Those perceptual quality measurement methods are generally composed of the following stages. First, a pre-processing step (scaling, color transformation and low-pass filter to imitate the point spread function of the eye optics). Next, a contrast sensitive function (CSF) filtering is applied. This allows to take into account the eye sensitivity to different spatial and temporal frequencies. The following step is a channel decomposition. Studies show that it is close to the behavior of the primary visual cortex. The last stages are error normalization and error pooling. Although considering some particularities of the HVS, those methods presented some

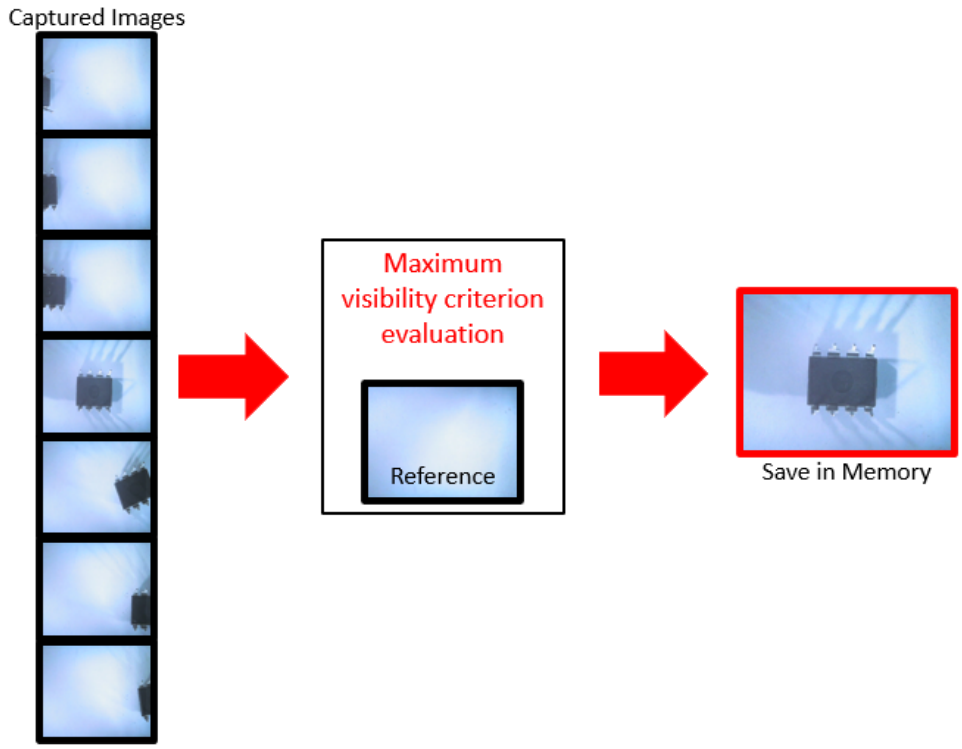


Figure 4.7: Illustration of image selection post-processing based on maximum visibility criterion

limitations. They are made of quasi-linear operators while the HVS is definitely more complex and nonlinear. The different limitations are pointed in [36]. The SSIM method overcomes these by assuming that human visual system is mainly retrieving structural information from a picture. The structural similarity index is computed combining a luminance, a contrast and a structure comparison as shown in figure 4.8. The structure comparison uses normalized signals in contrast and luminance: $\frac{(\mathbf{x}-\mu_x)}{\sigma_x}$ and $\frac{(\mathbf{y}-\mu_y)}{\sigma_y}$ where $\mathbf{x}$ and $\mathbf{y}$ are the original signals, μ_x and μ_y their respective mean intensity and σ_x and σ_y the standard deviations representing the contrast in each image. Those normalizations are justified on the basis that an image structure is independent of the average luminance and contrast. For instance, the pixel luminance is determined by the reflectance of the observed object as well as the illumination. This illumination is not affecting the structural information of the scene and should therefore be disregarded. The normalizations are defined locally since the contrast and luminance may vary inside a single image. The luminance comparison block is using μ_x and μ_y while the contrast comparison is based on σ_x and σ_y. The exact formulas for each of the comparison functions and the combination of those three are further detailed in [36].

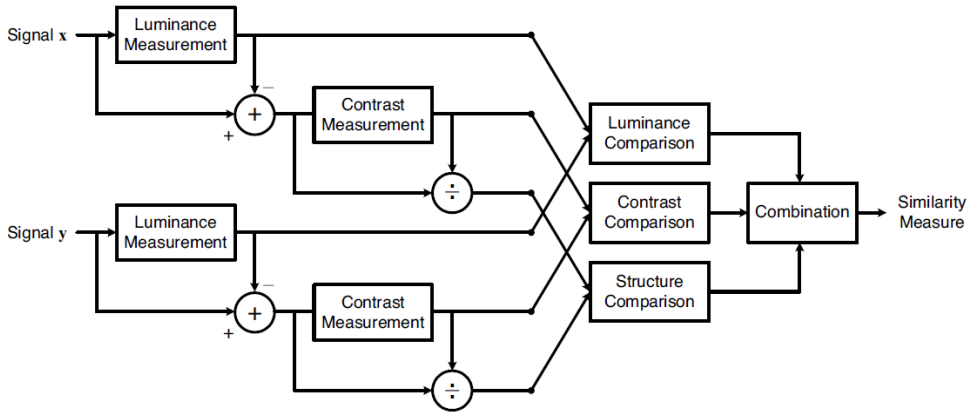


Figure 4.8: Diagram of the structural similarity (SSIM) measurement system. [36]

Weighted Mean Squared Error 1 (WMSE1)

In our application, it can be useful to give more weights to particular pixels inside the image array. More precisely, it would be a good practice to select an image where the insect is entirely visible and in the center of the picture. To this end, we can use the MSE definition given in 4.4 and multiply the central pixels by a weight factor. It is illustrated in figure 4.9, a score is computed for every regions of the images. When a pixel belongs to a central region (green in the image), the score is computed by doubling the MSE of this region. For peripheral pixels (blue), the score is the normal MSE. At the end, we can obtain a score for the whole image by taking the mean score of each regions. The image of the set with the greatest score presents a high visibility while being centered.

Weighted Mean Squared Error 2 (WMSE2)

Additionally to the method just presented, a variant can be proposed with a different weight distribution as shown in figure 4.10

4.4.2 Comparison of the different methods

To determine which of those methods is the most suitable for the insect monitoring application, some evaluations were performed using a data set generated to reproduce the inputs the post-processing methods could have to deal with. This data set, represented

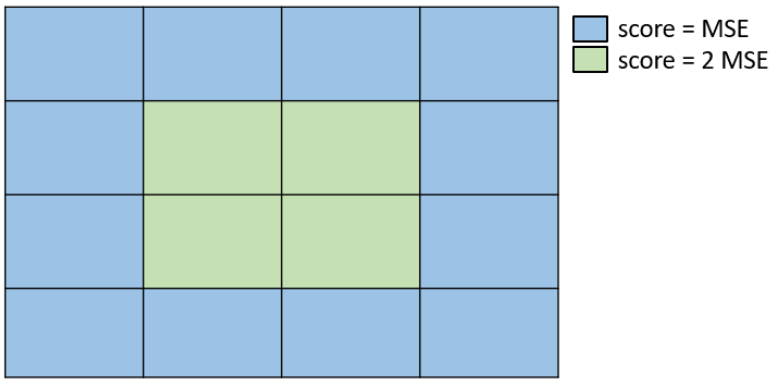


Figure 4.9: Spatial weight distribution for an image with the WMSE1 criterion

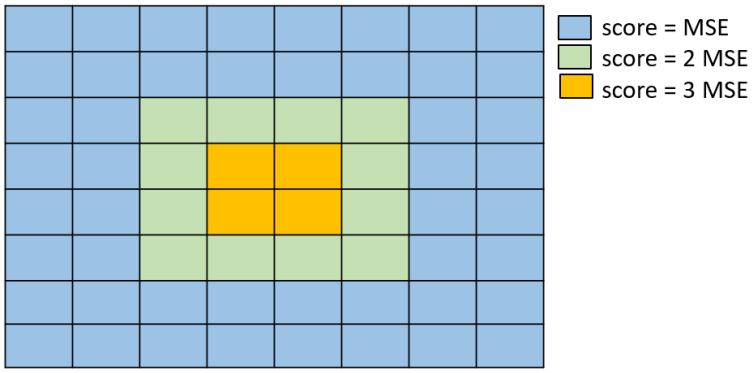


Figure 4.10: Spatial weight distribution for an image with the WMSE2 criterion

in figure 4.11, is a sequence of pictures of an electronic chip. The chip position is different on each picture. For the insect monitoring application, the post-processing method should select a picture where the chip is completely visible (picture 6 and 7 in the image set of Figure 4.11) and even better a centered picture (picture 7). The following evaluations determine whether each method is correctly selecting one of the most visible images among the set under different perturbations. Those perturbations are variations of luminance and addition of random noise. Within a test, evaluation is repeated 100 times with perturbations taking a random value in an interval specified in the test. The perturbation can either be constant over the image set or vary from images to images.

Execution time of the different methods

The first comparison to make between the selection methods is their execution time. Indeed, the post-processing phase should be as fast as possible to reduce the power consumption and increase the rate at which insect can be pictured. The different image

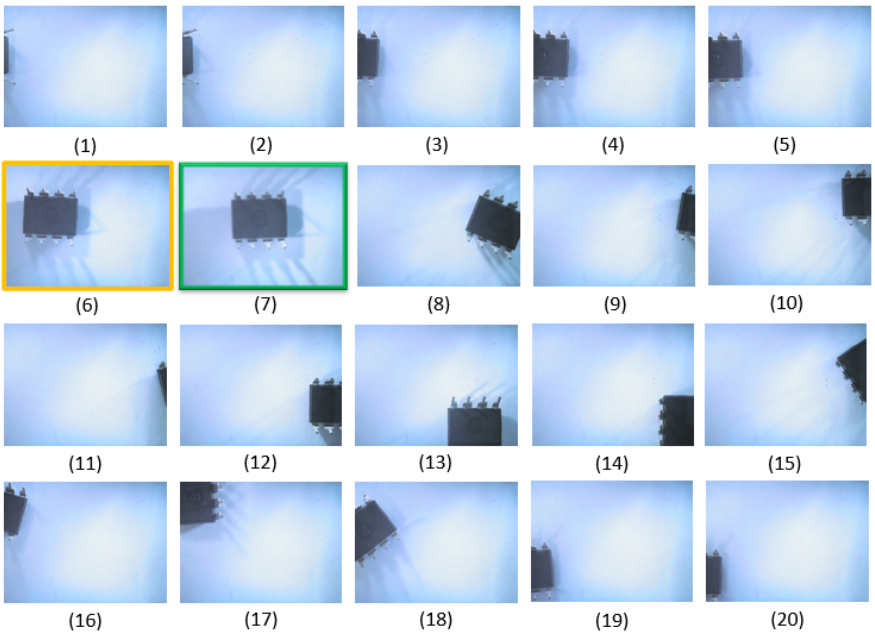


Figure 4.11: Image set used for comparison of selection methods

selection methods have been executed for 20 consecutive set of images, the mean execution times obtained for each of these are presented in Table 4.1. SSIM algorithm is much slower than the three others.

T_{MSE}	3.37 $[s]$
T_{SSIM}	50.63 $[s]$
T_{WMSE1}	2.66 $[s]$
T_{WMSE2}	3.2 $[s]$

Table 4.1: Time comparison for the post-processing selection methods

Luminance perturbation

For this test, a perturbation of luminance is applied to the image set shown in figure 4.11. The perturbation is a random integer taking a value in the interval $\{-x\ ;\ +x\}$, where x is the maximum luminance perturbation (increasing during the test). The effect of modifying the image luminance can be observed in Figure 4.12.

The selection of the best image (maximum visibility criterion) is evaluated for each of the methods presented before (MSE, SSIM, WMSE1 and WMSE2). The evaluation

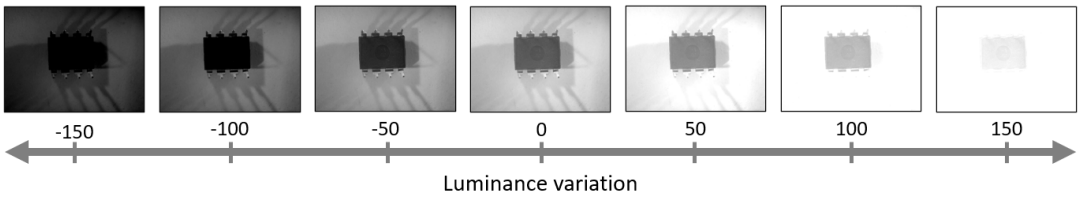
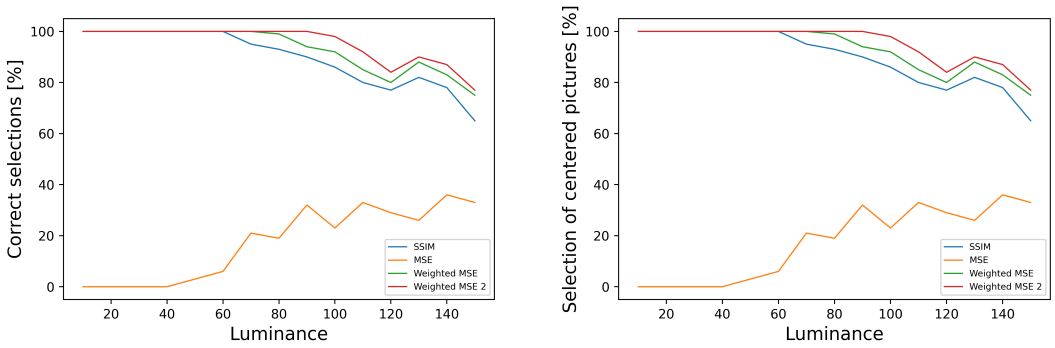


Figure 4.12: Luminance variation applied to an image of the data set in figure 4.11

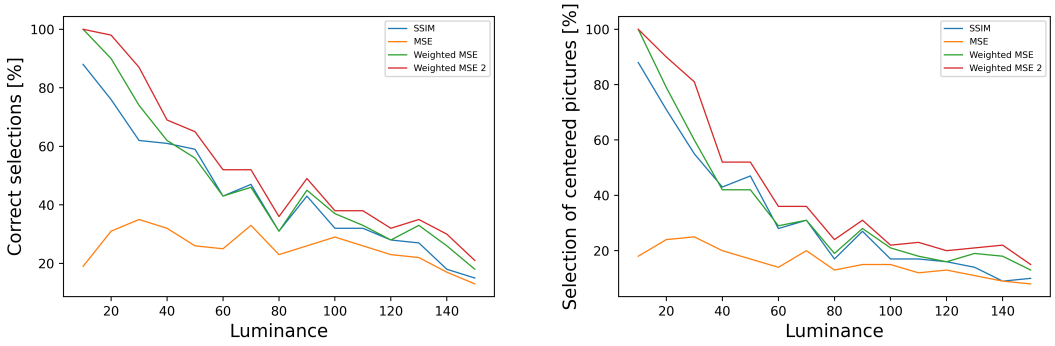
is repeated 100 times for each interval value $x = \{0\ ;\ 150\}$. At first, we evaluate the selection algorithms when the same perturbation (randomly taken in $\{-x\ ;\ +x\}$) is applied to the whole image set. The obtained results are highlighted in Figure 4.13. Figure 4.13a represents the percentage of correct selection (i.e. outputting picture 6 or 7 from the image set in figure 4.11) for each of the presented algorithm under increasing luminance perturbation. Figure 4.13b shows the percentage of tests where the selected picture was centered (picture 7 in the data set). The WMSE2 method performs slightly better than the WMSE1 and SSIM ones. Under mild perturbation (luminance level: $\{0\ ;\ 50\}$), those three methods are selecting an image where the object is completely visible and centered in 100% of the tests. The accuracy of selection drops around 80% for WMSE2 under luminance perturbations up to 150. Figures 4.13a and 4.13b present exactly the same results. It means that throughout the tests, picture 6 (completely visible but not centered) was never selected. MSE method presents bad selection results even at low luminance variation. This means that this algorithm is not suitable for the application.



(a) (b)

Figure 4.13: Comparison of image selection algorithm results under increasing luminance perturbation (same perturbation for all images in the data set). (a) Percentage of selected pictures where the chip is completely visible. (b) Percentage of selected pictures where the chip is centered.

Figure 4.14 presents the results of the different selection algorithms under luminance perturbation. This time, the amount of luminance added is not constant over the whole image set, a new value of perturbation is computed for each image. As can be observed, the performance of the algorithms is lower than for the previous experiment. Indeed, the algorithms are more likely to select wrongly when the perturbation is varying from image to image (some images are more affected than others). However, the WMSE2 method is still the best one and selects correctly under small perturbations (more than 80% of correct and centered selections for luminance perturbations $\in \{0\ ;\ 40\}$).



(a) (b)

Figure 4.14: Comparison of image selection algorithm results under increasing luminance perturbation (perturbation different for each image in the data set). (a) Percentage of selected pictures where the chip is completely visible. (b) Percentage of selected pictures where the chip is centered.

Random noise perturbation

This experiment takes the image set in Figure 4.11 and adds a random generated matrix (same size of the image). As for the luminance perturbation experiment, the noise level added to each pixel is increased gradually. Such perturbation can be observed in Figure 4.15. The results for selection algorithms under increasing noise level are shown in Figure 4.16 (same noise applied to the whole image set) and Figure 4.17 (different noise applied to each image of the set). In both cases the algorithms WMSE1 and WMSE2 are selecting perfectly under all noise levels. This may be because random perturbation at pixel level do not alter the global trend in the picture. Furthermore, it is applied to every picture so none is preferred to another. The algorithms SSIM and MSE performs badly at low noise level and good at higher perturbations.

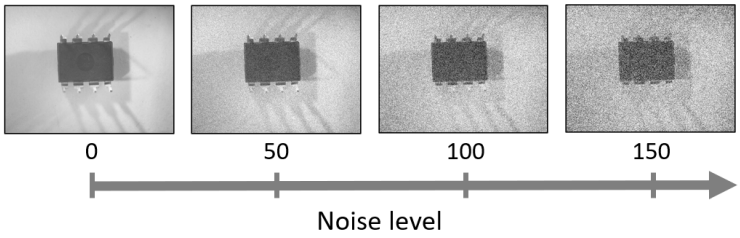
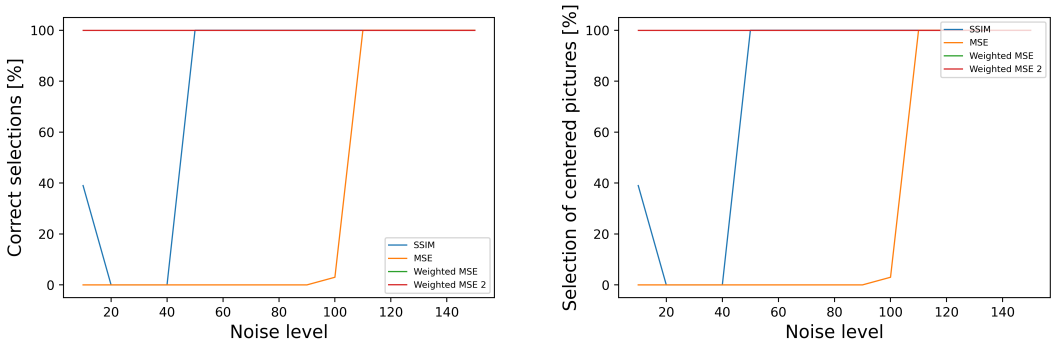
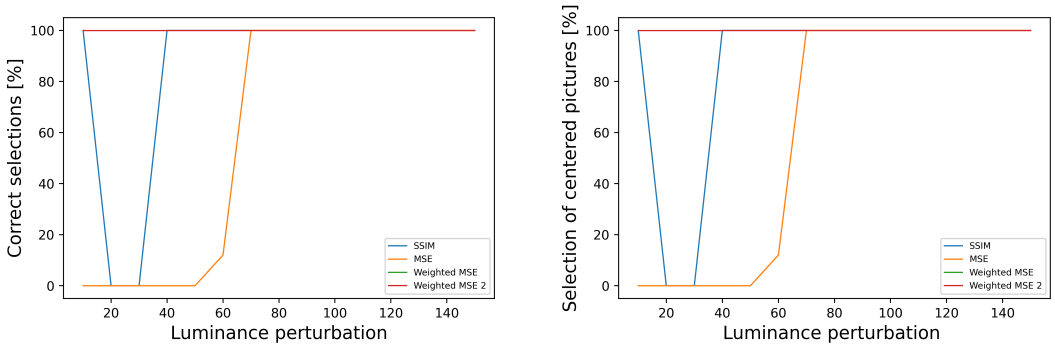


Figure 4.15: Random noise variation applied to an image of the data set in figure 4.11



(a) (b)

Figure 4.16: Comparison of image selection algorithm results under increasing noise level (same perturbation for all images in the data set). (a) Percentage of selected pictures where the chip is completely visible. (b) Percentage of selected pictures where the chip is centered.



(a) (b)

Figure 4.17: Comparison of image selection algorithm results under increasing noise level (perturbation different for each image in the data set). (a) Percentage of selected pictures where the chip is completely visible. (b) Percentage of selected pictures where the chip is centered.

Chapter 5

Test and assessment of the proposed design

The hardware and software considerations have been presented in chapter 3 and 4. This chapter aims at validating the proposed design and identifying potential limitations. First, section 5.1 evaluates the power consumption of the device during its different phases of operation. The operating time of the insect monitoring device is deduced from it. Section 5.2 computes then the image storage capacity of the microcontroller memory. After, section 5.3 presents the different outcomes from the outdoor experiments performed to test the device in real conditions. Finally, section 5.4 contains a summary table of the imaging device performance.

5.1 Power consumption

In this section, the power consumption of the different components constituting the monitoring device is evaluated. When possible, the expected and measured power are compared. Finally, the total consumption is estimated, allowing to determine the operating time of the device.

5.1.1 IR Sensor board consumption

Expected The datasheet tells us that the supply current is maximum 10 $[mA]$. Connected to a 3.3 $[V]$ voltage supply, this would correspond to a power consumption of maximum 33 $[mW]$.

Measured A first simple way to measure the power consumption of the sensor board is to measure the current taken from the Raspberry pins. The corresponding experimental bench and the connections for voltmeter and ammeter are presented in figure 5.1. The

current measured is 7 $[mA]$ and the voltage is not perfectly 3.3 $[V]$ but rather 3.07 $[V]$. This results in a power consumption of $P = VI = 21.49$ $[mW]$. This consumption is lower than the expected one because the latter was using a maximum value for the drawn current.

Additionally, we can estimate the power consumption of each component of the sensor

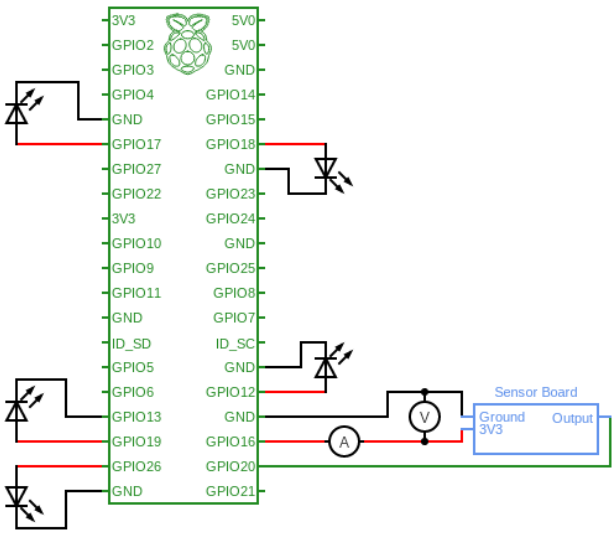


Figure 5.1: Experimental bench for sensor board power measurement

board circuit. The voltages measured at the different nodes of the circuit are indicated in figure 5.2.

The power dissipated through the resistors is given by :

$$\begin{aligned} P_{res} &= \sum_{i=1}^{5} \frac{V_i^2}{R_i} \\ &= \frac{(3.07-1.533)^2}{22k} + \frac{(3.07-1.126)^2}{300} + \frac{(3.07-1.324)^2}{10k} + \frac{(3.059-2.609)^2}{1k} + \frac{1.533^2}{22k} \\ &= 13.32\ [mW] \end{aligned}$$

where V_i is the voltage across the resistor R_i. The power dissipated in LED 1 can be computed with the current through it (and through R_4 too):

$$P_{L_1} = VI = 2,609\ \frac{(3.059-2.609)}{1k} = 1.17\ [mW]$$

Similarly, the power consumption of the infrared LED of the sensor is :

$$P_{LED,sensor} = 1.126\ I_{R_2} = 1.126\ \frac{(3.07-1.126)}{300} = 7.3\ [mW]$$

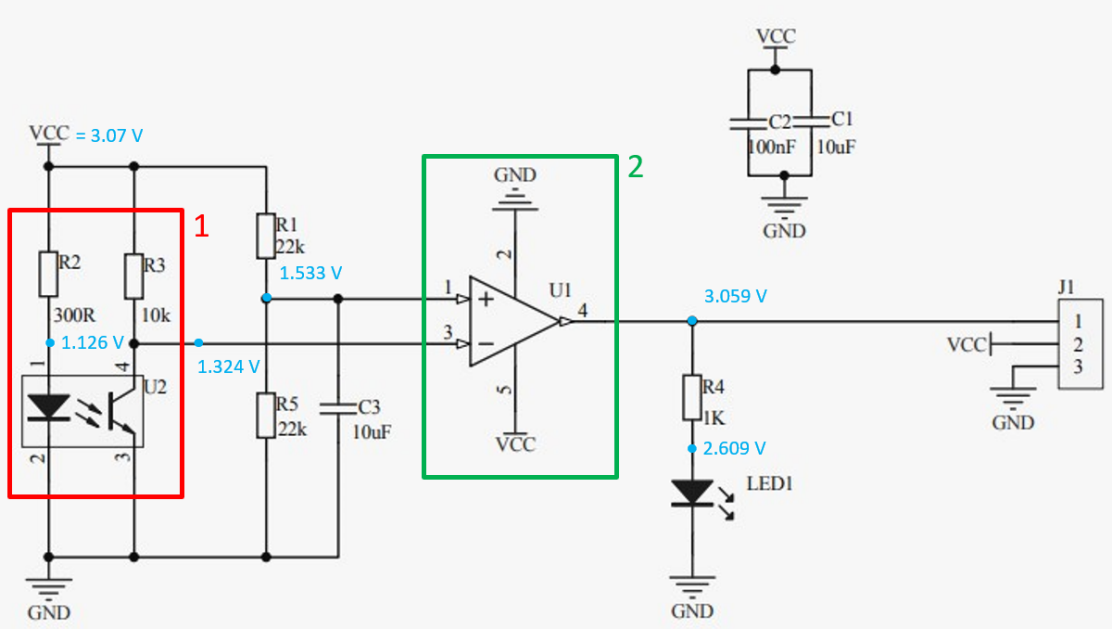


Figure 5.2: Voltage measurement on the sensor board circuit

We can observe that the most consuming components are the resistors and the IR LED of the photoreflective sensor.

5.1.2 LED's consumption

Expected The LEDs used in the design are 5mm white LEDs whose current is 20 $[mA]$ when connected to 3.3 $[V]$ supply. This gives a power consumption of 66 $[mW]$ for each and a total consumption of $P_{LED,theory} = 330$ $[mW]$ for the 5 of them.

Measured To estimate the power consumed by the LEDs ensuring the lighting inside the trap, we can measure both the voltage and the current flowing through those. For one LED, the current measured is in average 9.5 $[mA]$ and the voltage is ≈ 3.03 $[V]$. This gives a power of $P_{LED} = VI = 28.785$ $[mW]$. The total power consumption associated to the LEDs of the system is simply the product of this value by the number of LEDs. In the final design, there are 5 LEDs. We have then $P_{LED,tot} = 143.925$ $[mW]$. Here again, the power measured is lower than the one predicted with data sheet information because both the voltage and the current is lower.

5.1.3 Camera consumption

The specifications of the camera states that it has a power consumption of 2.5W. This means that the total power consumed by the cameras is $P_{cam} = 7.5\ [W]$. There is no way to directly measure the power consumption of the camera.

5.1.4 Processor consumption

In [37], we learn that the Raspberry Pi model 3b+ is consuming approximately 2.295 $[W]$ when idle and up to 5.661 $[W]$ when the processor is performing heavier computations. There is no way to measure the power consumption of the microcontroller only.

5.1.5 Total consumption

Expected From the subsections presented before, we can gather the expectation information from each subpart to estimate the total power consumption of the monitoring device. We will distinguish the consumption when there is an insect detected or not. In the first case, the power consumption will be maximal: the sensor board, the cameras and the lighting LEDs are ON and we can assume that the raspberry Pi is processing lots of information. The expected maximal power consumption is therefore:

$$\begin{aligned} P_{theory,max} &= P_{sensor} + P_{LED,tot} + P_{PI,max} + P_{cam} \\ &= 0.033 + 0.33 + 5.661 + 7.5 = 13.524\ [W] \end{aligned} \tag{5.1}$$

The minimal power is obtained in between detection events because only the sensor board is powered, the processing of the raspberry Pi can be considered low as well.

$$\begin{aligned} P_{theory,min} &= P_{sensor} + P_{PI,idle} \\ &= 0.033 + 2.295 = 2.328\ [W] \end{aligned} \tag{5.2}$$

These considerations help to predict a range for the power consumption : $[2.328 - 13.524\ W]$. The actual power consumption is unlikely to be correctly estimated by those two boundaries. Indeed, the power consumption of the processor is highly uncertain during operation and the assumptions taken to compute the power during idle and image acquisition modes are probably not valid. For this reason, the following developments, based on measurements in real conditions, aims at providing a better estimate of the power consumption during the different phases of activity of the monitoring device.

Measured When the device is in operation, we can distinguish different activity modes. Those modes will have different power requirements and should therefore be taken into account for the total power consumption. The different activity modes are the following:

1. **Idle mode**: it is the principal mode. During this mode, the Raspberry Pi is waiting for a triggering signal from the sensor. Power is only consumed by the sensor board and the raspberry Pi. As the latter is not executing user processes, the power consumption is low. The current drawn from the battery is 41 $[mA]$ in average. The average power measured for this mode is then:

P_{idle}	2.05 $[W]$

2. **Image acquisition mode**: This mode is activated once the sensor has detected the presence of an insect. It is the most power-consuming one since it is activating the cameras and the Raspberry Pi consumption also increases while managing the capturing images.

$P_{acquisition}$	5.425 $[W]$
$T_{acquisition}$	5.2 $[s]$

3. **Post-processing mode**: As already mentioned, once the cameras have finished capturing images, a post-processing step is required to filter the obtained pictures and keep only the best picture in memory (WMSE2 method was selected). This process is associated to higher computational effort than during idle mode and for this reason, it is important to consider it as a mode in itself. The average power and time measured are :

$P_{post-process}$	2.9 $[W]$
$T_{post-process}$	3.2 $[s]$

Intuitively, the total power consumption will be dependent on the efficiency of the upstream insect trap and the amount of insects that will enter our device. Indeed the image acquisition and post-processing modes are activated at each insect visit. Those modes being more power-consuming than the idle mode, the operating time will decrease with the number of insect observed. Figure 5.3 illustrates the total power consumption of the monitoring device as a function of time. This graph is not associated to real experiments but aims at providing an estimate of the actual consumption when detection occurs (red dashed line in figure 5.3).
When several insects are detected, the portion of time where power is high increases. We can therefore define the equivalent power consumed as a function of the number of insects passing through the device per hour.

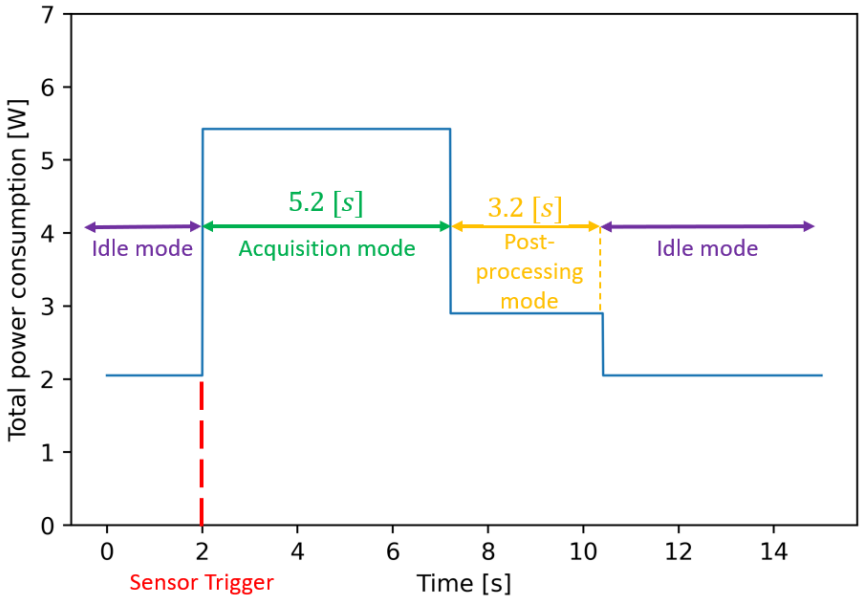


Figure 5.3: Total power consumption within the different mode of operation

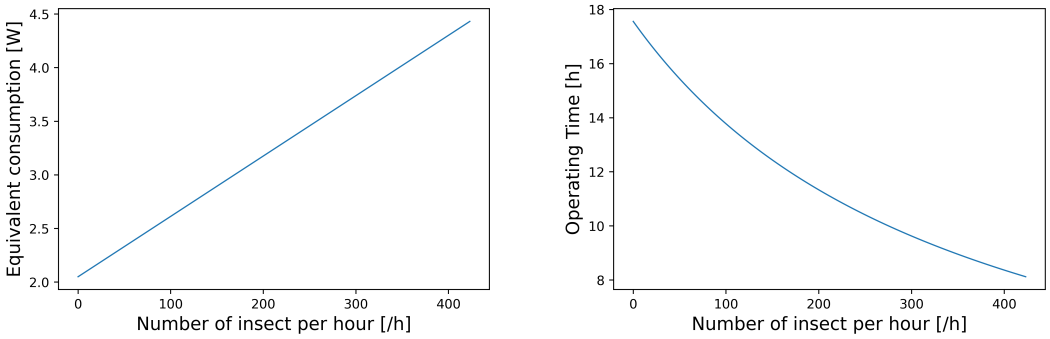
$$P_{eq} = P_{idle}\frac{3600 - x\,(T_{acquisition} + T_{post-process})}{3600} + P_{acquisition}\frac{x\,T_{acquisition}}{3600} + P_{post-process}\frac{x\,T_{post-process}}{3600} \tag{5.3}$$

where x is the number of insects per hour.
The evolution of the equivalent power consumption is shown in Figure 5.4a. It has been evaluated that the image acquisition and post-processing modes last 5.2 $[s]$ and 3.2 $[s]$ respectively, leading to a total duration of 8.4 $[s]$ for each insect observation. For this reason, the number of insects per hour considered in Figure 5.4a is ranging from 0 to 423. The upper boundary corresponds to a visit of insects every ≈ 8.5 $[s]$. The operating time of the monitoring device can be deduced from the equivalent power and the battery energy capacity. The battery used for this project has a capacity of 36 $[Wh]$ (Appendix A). The operating time is therefore given by :

$$T_{operating} = \frac{36}{P_{eq}} \tag{5.4}$$

The resulting operating time as a function of the number of insect visits per hour is illustrated in Figure 5.4b. Very intuitively, the lowest power consumption and longest operating time are observed in the case when no insects enter the trap. In this situation, the equivalent power is equal to $P_{idle} = 2.05$ $[W]$ and the corresponding operating time is 17.56 $[h]$. In the worst case, insects are entering the trap at a rate where there is no

idle mode (a new detection occurs right after the post-processing phase is ended). This corresponds to ≈ 423 insects per hour. Then, the equivalent power consumption is equal to 4.43 $[W]$ and the operating time drops to 8.12 $[h]$.



(a) (b)

Figure 5.4: (a) Evolution of the equivalent power consumption of the monitoring device with increasing amount of insect visits per hour (as predicted by Equation 5.3) (b) Evolution of the operating time of the monitoring device under increasing amount of insect visits per hour

Operating time assessment

As validation to this operating time prediction, the time to completely discharge the battery when the code is running has been measured. During the experiment, the monitoring device is taking a picture every 10 minutes. This would correspond to an equivalent power of 2.11 $[W]$ using 5.3. The predicted operating time is then 17h05min. The experiment was started at 8:38 am and the battery was empty around 11:28 pm (last picture recorded). The measured operating time is thus 14h50. This is slightly lower than the predicted duration. This difference can be explained by the fact that throughout the validation experiment, the Raspberry Pi was connected to an external computer with a Virtual Network Computing (VNC) connection. This is responsible for a higher energy consumption.

5.2 Image storage capacity

The memory size for one picture was already computed in Equation 4.1. Although, the actual size used on the Raspberry pi is greatly reduced by the JPEG encoding. The storage required for one picture is ≈ 68 kB rather than the 921.6 kB computed before. The full Raspberry Pi Operating system (OS) packages need 8.61 $[GB]$ storage [38]. A

reasonable assumption would be to consider that the total unusable storage is 10 $[GB]$. Taking a 16 $[GB]$ SD-card, that leaves 6 $[GB]$ memory space for images. This corresponds to $\approx$ 88235 images.

5.3 Outdoor testing

5.3.1 Image selection algorithm assessment

As was proven in 4.4.2, perturbation in ambient luminance can lead to wrong selection of the most visible image during post-processing. This section aims at characterizing the luminance variation that could occur in real conditions.
The monitoring device has been installed outside (position: New Haven, Connecticut (CT) United States of America (USA) (41.294, -72.896), orientation: North-West) from 9:08 am to 11:28 pm, taking pictures every 10 minutes (total of 88 images captured). I placed a bud in front of the cameras to have a non-empty image. The object position is constant along the whole experiment so that the only variation observed is the ambient illumination.
The mean luminance of the images measured over time is shown in Figure 5.5 for the top view camera. The brighter image is obtained around 3:58 pm where the average luminance is approximately 220. The measured maximum and minimum luminance for each of the 3 views are written in table 5.1. The standard deviation along the day was also computed. The top view shows the highest luminance variation among the cameras with a standard deviation of 29.75 and a difference between its maximum and minimum value of 82.51. Referring to the experiment in subsection 4.4.2, such luminance variations lead to poor image selection and we risk to save useless insect images in memory (incomplete, poor visibility). A simple solution can be implemented to mitigate this problem: capturing reference images again at regular intervals. Indeed, wrong selection is the consequence of the comparison of an image set whose global illumination has been shifted (due to ambient light) to a fixed reference. If we renew the reference image (every hour e.g.), the luminance variation will be reduced and a better selection is possible. Figure 5.6 shows the results of the experience described in subsection 4.4.2 when the reference is perturbed by the same amount of luminance. It can be observed that the performance is increased. As a conclusion, renewing the reference at regular time intervals is beneficial for the post-processing selection method.

5.3.2 Experimental methods

The developed insect monitoring system has been tested in real conditions 7 times during daytime for durations varying from 2h30 to 8h (depending on weather conditions). All

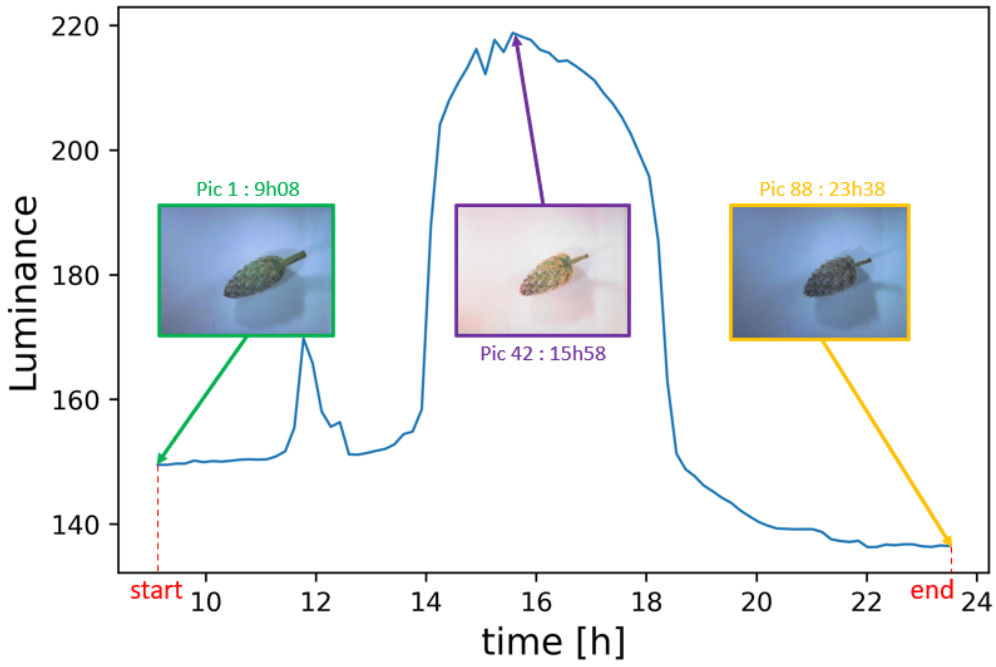
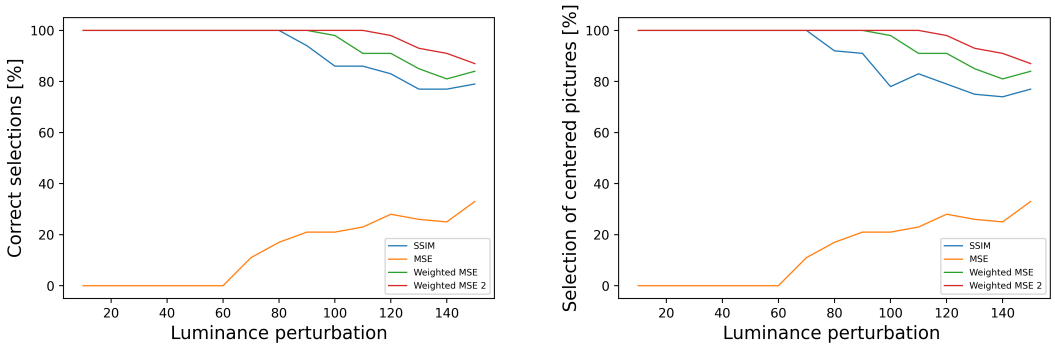


Figure 5.5: Luminance variation with time under real conditions

Camera	Minimum	Maximum	Mean	Standard deviation
Side view	104.1	178.76	127.45	27
Below view	82.8	152.21	101.29	25.5
Top view	136.32	218.83	165.21	29.75

Table 5.1: Mean, minimum, maximum and standard deviation of the luminance measurements for each camera view

the experiences were repeated around the same location (Marsh Botanical Gardens, New Haven CT, USA (41.321, -72.924)). The trap was installed following the guidelines in [8], i.e. at a location in between flowered areas (flight corridor). More detailed information related to the testing conditions of the different experiments can be found in Appendix D. The 6 first tests did not yield any results in terms of insect observation. The reality of outdoor testing revealed some deviations in the trap functioning compared to in-lab experiments. For this reason, the next subsection is a report of the different iterations brought to the design to improve its operation.



(a) (b)

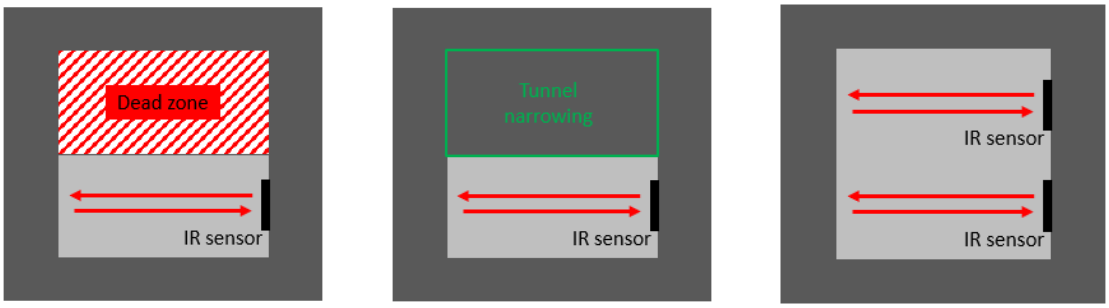
Figure 5.6: Comparison of image selection algorithm results under increasing luminance perturbation (reference perturbed accordingly). (a) Percentage of selected pictures where the chip is completely visible. (b) Percentage of selected pictures where the chip is centered.

5.3.3 Iterating the design

Once outside, the design presented in section 3.3.1 showed some non-idealities not predicted during in-lab characterization. This subsection is summarizing the different steps of the evolution of the design towards a more robust and efficient device for outdoor monitoring of insects.

Flexible connection to the malaise trap When the trap is installed, the tension in the nets is impacting the connection in between the collecting bottle of the malaise trap and the entrance of the conceived device. During test 2 (Appendix D), the installation led to the breaking of the interface piece due to those efforts (shear stress). To prevent this from happening again, I added some compliance in the connection between the trap and the monitoring assembly by making the interface piece flexible (the piece presented in Figure 3.24a is already the flexible version, previous version was 3D-printed).

Narrowing the path in front of the sensor As can be observed in Figure 3.20, the sensor is not covering the totality of the device entrance. When observing the behavior of insects walking in the trap, it has been noticed that the insects do not always walk on the bottom part of the entrance, they can also walk on the walls. Therefore, they could walk over the IR sensor without being detected (dead zone out of the sensed field, see in Figure 5.7a). A simple solution to this problem is to restrict the entrance path to the height of the sensing area (Figure 5.7b). Another possibility is to use a second IR sensor to cover the dead zone left by the first one (Figure 5.7c).



(a) (b) (c)

Figure 5.7: (a) Actual coverage of the entrance by the IR sensor. (b) Solution 1: reduce the height of the entrance. (c) Solution 2: use a second IR sensor.

Shielding from ambient light During test 2 (Appendix D), it was noticed that the detection was not working as efficiently as inside the lab. To understand this phenomenon, the voltage at the output of the sensor (collector of the phototransitor) in the two different conditions was measured:

Output voltage inside lab	1.345 $[V]$
Output voltage outdoor[1]	0.156 $[V]$

This difference in voltage is due to the amount of IR radiations perceived by the phototransistor. As the voltage measured outside is higher than the one recorded inside the lab, it means that the sensor receives more IR radiations when exposed to sunlight. The Sun can be considered as blackbody whose temperature is 6000K. This concept was presented generally in section 2.1. The corresponding spectral irradiance is shown in figure 5.8 (green). The actual solar radiations perceived on Earth's surface differ from this curve. When light travels from the Sun to Earth, it encounters a number of obstacles. Those are affecting the overall power density reaching Earth's surface but also its spectral content (the impacts is greater on some wavelengths). First, light needs to go through Earth's atmosphere. While traversing it, photons can be absorbed or scattered leading to a reduction of the power density and variations of the direction of light radiations [13]. Then, photons can also be impacted by pollution, clouds or water vapor. In figure 5.8, the green curve is the sunlight intensity on Earth's surface predicted by the blackbody theory. To take into account the effect of atmosphere, we need to consider the length traveled by light inside it (longer path implies more interactions with particles). The Air Mass (AM) is a quantity characterizing this path length. It is defined as the actual path light is passing through in the atmosphere normalized by the smallest path possible (sun

[1] The outdoor conditions were sunny with some clouds

perfectly overhead):

$$AM = \frac{1}{\cos\theta}$$

where θ is angle between the vertical and the direction of the sunlight. This quantity is used to define standardized solar spectrum on Earth surface. In red in 5.8, the AM0 is the spectral irradiance that would be observed if there was no atmosphere. In blue, it is a most common spectrum : AM1.5. It can be observed on this curve that it presents deep troughs at some wavelengths. The reason behind that is the gas constitution of the atmosphere, photons are highly absorbed by gasses whose bond energy is equal to the photons one. The gasses responsible for the different deep troughs are highlighted on the graph as well (ozone O_3, carbon dioxide CO_2 and water vapor H_2O).

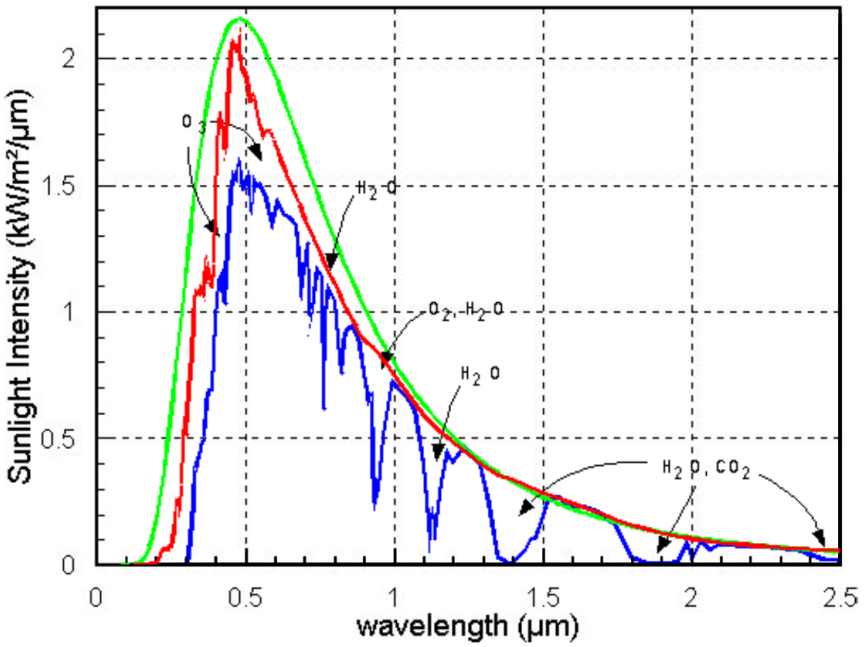


Figure 5.8: Comparison of solar radiation outside the Earth's atmosphere with the amount of solar radiation reaching the Earth itself [13].

The visible light ranges from 380 $[nm]$ to 750 $[nm]$, IR occupying longer wavelengths (750 $[nm]$ to 100 $[\mu m]$) [14]. The IR photoreflective sensor SEN0017 (Appendix A) has its peak emission around 940 $[nm]$ [28]. It can be deduced from Figure 5.9 that the sunlight spectrum has a much greater component at this wavelength than artificial lighting sources (except for incandescent lights) [39]. This explains why the experience presented in section 3.2.3 (Light influence) did not express the necessity of any shielding from ambient light. The light source used was the LED flash light of a OnePlus 3T smartphone.

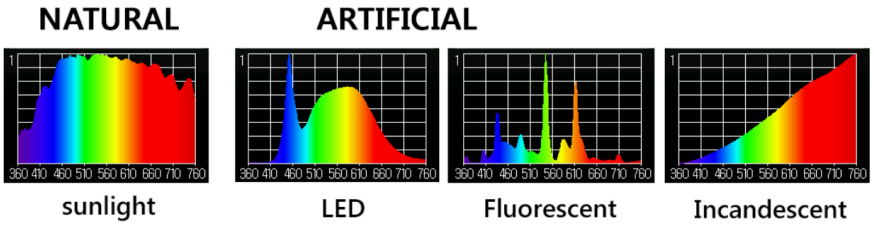


Figure 5.9: Comparison of natural and artificial light spectrums [39]

To increase the robustness of the device to varying ambient light conditions, some shielding solutions were adopted. First, the box was covered with a black towel. The measured output voltage of the IR sensor was then 0.26 $[V]$. It is slightly better than the 0.156 $[V]$ measured without cover. The different plates constituting the device assembly have been painted in black. The voltage measured at the output of the sensor after painting is presented in table 5.2. The sensor being located near the entrance of the monitoring device, it is exposed to more light. When the interface piece is connected (Figure 3.24a), it protects partially the IR receiver from light radiations. It can be observed that the black painting by itself did not improve the IR shielding a lot. Although, when combined with an additional black towel covering the box, the measured voltage reaches 0.78 $[V]$ and even 0.985 $[V]$ when the interface piece is connected. The goal is to get a voltage in the same range as the one measured in laboratory conditions (1.345 $[V]$).

Interface piece connected	0.26 $[V]$
Interface piece connected and covered	0.985 $[V]$
No interface piece connected	0.21 $[V]$
No interface piece connected but covered	0.78 $[V]$

Table 5.2: Measured voltage on the IR sensor when painted in black

5.3.4 Results : insect observation

Appendix D groups the different outdoor experiments and the associated observations. Insect pictures were only taken during test 7. Although, previous tests reveals interesting details. First, they allowed to refine the design as explained in subsection 5.3.3. In tests 3-7, insects were present upstream the imaging box (Figures D.3, D.4 and D.5) but almost none after. Something in the path was wrongly designed for the insects to go through. First, as explained in [8], the malaise trap exploits the natural tendency of insects to fly upwards. However, the interface piece between the commercial trap and the imaging box implies that insects fly downwards once caught to go in the glass tunnel

(see figure 3.24b). Another observation from the different outdoor experiments is that insects avoid darker places. Using a black cover implies less illumination at the entrance of the imaging box which is good for the detection ability of the IR but detrimental to the insect circulation.
Finally, let us discuss the result obtained during test 7. No detection was recorded during the operation of the trap. When the trap was being uninstalled, it was noticed that the collecting bottle of the malaise trap was full of insects (Figure D.4). By removing the stakes of the trap, the exit of the trap (bottle) could stay connected to the monitoring device through the interface piece. In a few minutes, some insects went through the trap and could be pictured (see Figure 5.10). A first observation is that the global brightness is low, the lighting should therefore be increased. The below view pictures do not show insects completely. It was noticed that the first insect was still during imaging and that the IR sensor had been triggered by another insect. This explains why the selection algorithm outputted this image (the insect was at the same position on all pictures).

After this, the interface piece was left connected and the monitoring device was

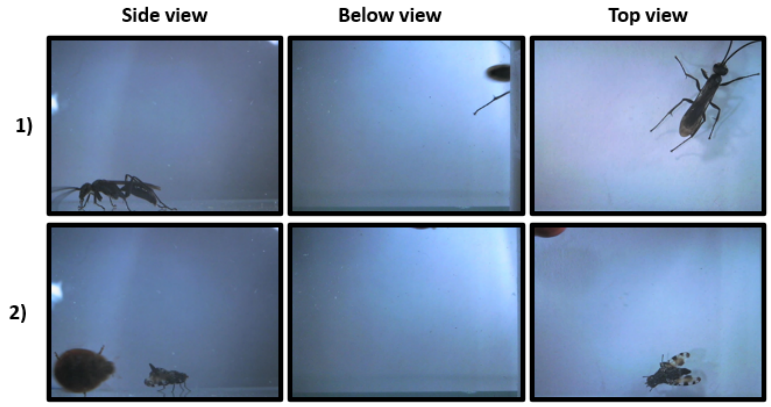


Figure 5.10: Observed insects during test 7

brought back to lab. The device kept running during 4 more hours, detecting 40 other events. Insects were present on some images. The rest of the detections must have been associated to insects going in the other direction (leaving the tunnel). Some insects may also head back in the bottle right after being detected without entering the glass tunnel. During those tests, the outside of the trap was closed with a collecting box to be able to verify the amount of insects detected. Leaving the exit open might reduce the number of insects going back in the wrong direction.

5.4 Final performance

As a conclusion to this validation chapter, Table 5.3 summarizes the overall performance of the designed monitoring device.

Power consumption	$P_{min} = 2.05\ [W]$ $P_{max} = 4.43\ [W]$
Operating time	$T_{min} = 8.12\ [h]$ $T_{max} = 17.56\ [h]$
Storage capacity	6 [GB] = 88 235 images
Detection capability	max 1 insect every $\approx 8.4\ [s]$
Image resolution	640x480
Weight	1.77 [kg]

Table 5.3: Performance of the monitoring device

Chapter 6

Discussion

This section discusses the outcome of the thesis, highlights its contributions compared to existing works, recognizes its limits and anticipates improvement directions.

The major contribution of this thesis is the design of an automated imaging device for insect monitoring that combines an IR sensor-based detection with camera imaging. It can operate autonomously for a daytime period (from 8.12 to 17.56 $[h]$ of autonomy). It has been designed using off the shelf components in an open-source and user-friendly perspective. This monitoring device was tested on a malaise trap but the principle could be extended to other existing insect traps and therefore target a broader variety of species. This would only require to design new interface parts (Figure 3.24a) to connect the output of the upstream trap to the entrance of the designed imaging box.

In opposition to the other automated monitoring solutions presented in section 2.3, this work also collects insect images. Indeed, in [17], [18], [7], [19] and [4], various sensors were used to count specimens or extract features such as their sizes but they were never combined with cameras.
In [21] conversely, a camera was used both for detection and imaging. The continuous operation of the camera results in a greater power consumption that could be avoided using another low-power sensor-based detection as it is the case in this thesis.

The proposed monitoring device presents some limitations. First, the device can only handle one insect at a time. In opposition, [20] and [21] use systems that captures images of an area that could be occupied by several specimens. [21] can track different individuals simultaneously.
When tested in real conditions, the designed imaging box showed a particular sensitivity to ambient conditions. This has a negative effect on detection. Improvements were obtained by shielding the sensor from sunlight (subsection 5.3.3) but there is still need for further progress in this direction.

The different tests with living insects also revealed that despite a large number of insects caught by the malaise trap, almost none went trough the imaging tunnel. It was identified that insects tend to fly upwards and the geometry of the interface part was inadequate for insect circulation towards the imaging box. Removing the bottle that was attached in the commercial malaise trap (Figure 3.24) and connect the converging point of the nets directly to the imaging box entrance could result in more insect observations as it would not require them to fly downwards.

A major drawback observed during test 7 (Appendix D) is the overall darkness of the recorded pictures. The current lighting inside the trap is ensured by 5 LEDs providing insufficient luminosity to the different camera views. The camera used in the imaging device are equipped with a built-in lighting. The latter was not currently used because its direct reflection on the glass tunnel prevents a clear observation of the inside of the tunnel. An opportunity for improvement would be to cancel this reflection to be able to use the built-in lighting of the cameras during imaging. This would benefit the brightness of the pictures but also their quality. If the lighting is sufficient, the exposure time can be decreased and with it, the risk of blurry insect images.

In subsection 5.3.1, it was evaluated that renewing the reference image regularly had a positive impact on the selection of the best image after image acquisition. This is a simple improvement perspective.

Finally, the current operating time is ranging from 8.12 to 17.56 $[h]$ which is sufficient for daytime recordings. Although, longer period measurements could provide useful statistics on insect abundance/diversity (seasonal, yearly time scales). Connecting the battery to a solar cell could lead to a much longer operating time. If the charging capacity of the solar cell is sufficient to ensure no interruption in operation, the limiting factor becomes the storage on the memory of the Raspberry PI (a manual collection of the stored images is necessary once it is full). This could also be avoided by adding a communication system to the monitoring device to send directly the captured images to a remote computer.

Chapter 7

Conclusion

The designed insect monitoring device satisfies the specifications mentioned in section 3.1. Indeed, it is a non-invasive solution capturing insect images from 3 different view angles (top, below and side). The imaging process is automated thanks to an IR photoreflective sensor located at the device entrance. Once insects have been detected and pictures taken, they can fly away freely through the trap exit.
In an effort to make this accessible to non-technical users, the device is made of off the shelf components whenever possible. The device was designed to interface a malaise trap but is modular and could be extended to other insect traps.
The power consumption of the device is relatively low ensuring an operating time ranging from 8.12 to 17.56 $[h]$ depending on the number of insect visits. The minimum time between two specimen observations is ≈ 8.4 $[s]$ which corresponds to a decent rate.
In addition, a selection code was implemented for post-processing the set of images captured after the IR sensor trigger. This code (WMSE2) aims at keeping only the image where the insect is the most visible. 100% of correct selection were observed under luminance and noise perturbations (same on all images from the set) ranging from 0 to 80 which is the level of variation measured in real conditions operation. The accuracy drops to $\approx 80\%$ when subjected to higher perturbations.
Chapter 6 finally presented the contributions and limitations of this work as well as possible ways of improvements. Although, the automated imaging device still presents some limitations, the outdoor testing identified promising improvement perspectives. Future work can help to better understand insects decline and trigger preservation actions.

Appendix A

Commercial products used in the design

Sensor Board - SEN0017, DFRobot

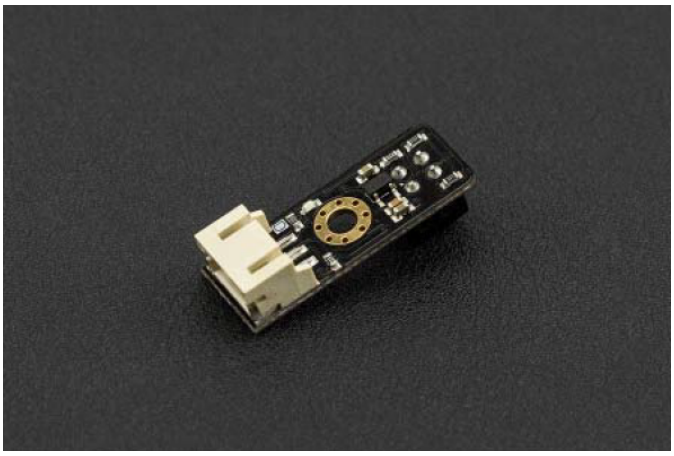


Figure A.1: SEN0017 from DFRobot

Link SEN0017 - Purchase link

Description

- Power supply: 3.3-5V, Operating current: <10mA
- Detecting Range: 1- 2cm
- Output interface: 3-wire interface (1 - signal, 2 - power, 3 - power supply negative)
- Schematic - Download

Cameras - Plugable USB 2.0 Digital Microscope

Figure A.2: Plugable USB 2.0 Digital Microscope

Link Plugable USB 2.0 Digital Microscope - Purchase link

Description

- High definition : 2.0 Megapixels, up to 250x magnification
- Integrated lighting: LED halo light with brightness adjustment control
- Image format: JPEG
- Video format: YUY2 (USB 3.0+ Only) or MJPEG
- Video capture resolution: 640x480 @ 30Hz
- Power: Bus Powered (No Power Adapter). Voltage: 5V, amperage: 500mA, wattage: 2.5W.

Microcontroller - Raspberry Pi 3 Model B+

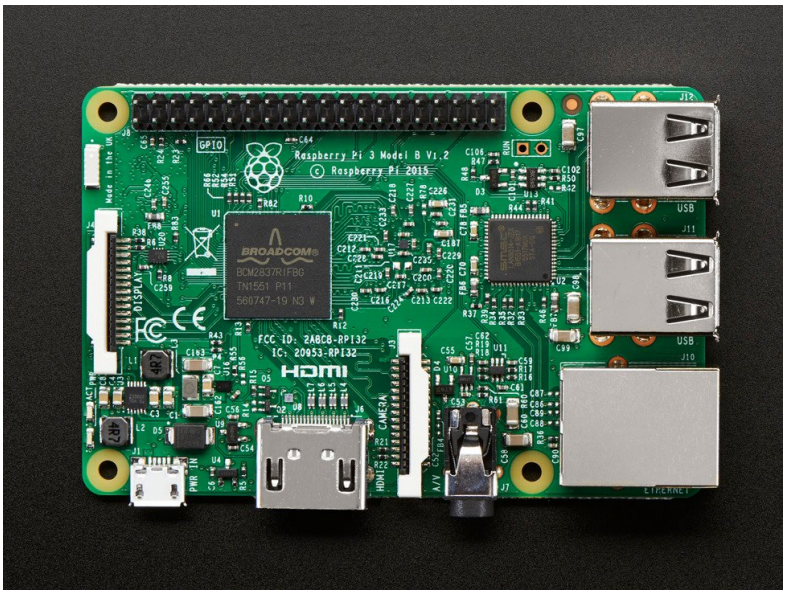


Figure A.3: Raspberry Pi 3 Model B

Link Raspberry Pi 3 Model B+ - Purchase link

Description

- Quad Core Broadcom BCM2837 64-bit ARMv8 processor. Processor speed up to 1.2GHz.
- BCM43438 WiFi chip
- Power consumption: 459 mA (2.295 W) at idle and up to 1.13 A (5.661 W) under full stress.
- 4 USB 2.0 ports, Gigabit Ethernet over USB 2.0, Full-size HDMI
- Extended 40-pin GPIO header
- Micro SD port for loading the operating system and storing data
- 5V/2.5A DC power input
- Dimensions: 86.9mm x 58.5mm x 19.1mm / 3.4" x 2.3" x 0.8"

LEDs

Figure A.4: LED (white)

Link LED - Purchase link

Description

- 5mm diameter
- Cool White color
- 3.0V Typical Forward Voltage, at 20mA current
- 15,000 mcd minimum brightness

Battery

Figure A.5: Battery

Link Battery - Purchase link

Description

- Lithium Ion
- 5 V
- 10,000 mAh/ 36Wh
- Weight: 180g

Malaise Trap

Figure A.6: Townes-style Malaise trap

Link Malaise Trap - Purchase link

Description

- Black Polyester no-see-um fabric (96 x 26 mesh/square inch).
- Interception area (center panel): 165 by 110 cm.
- Equipped with one 500 ml collecting bottle.
- Tent pegs, guy ropes, and support poles required for trap installation are not included in the package.

Screws and nuts

Links McMaster-Carr - screws and McMaster-Carr - nuts

Part references Table A.1 groups the references for all the screws and nuts required for the device assembly.

Description	Reference
18-8 Stainless Steel Socket Head Screws M3 L6	90751A110
18-8 Stainless Steel Socket Head Screws M3 L12	90751A113
18-8 Stainless Steel Socket Head Screws M2.5 L6	91292A010
Metric Medium-Strength Steel Hex Nuts M3 - Class 8	90592A085
Metric Medium-Strength Steel Hex Nuts M2.5 - CLass 8	90592A080

Table A.1: References for screws and nuts purchase

Appendix B

Technical drawings - Structure plates

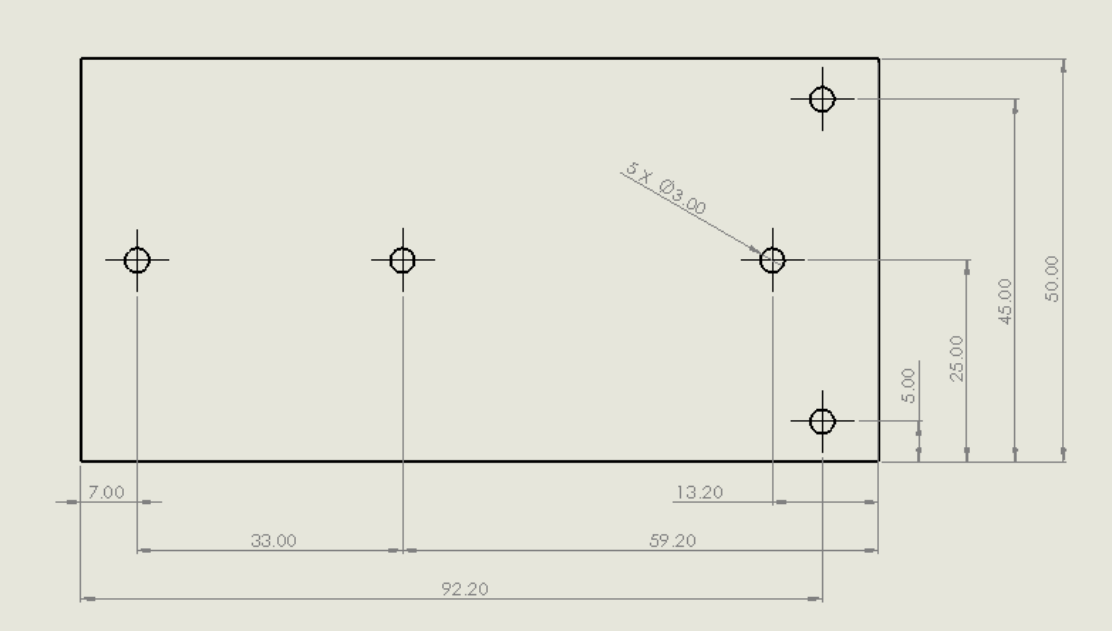


Figure B.1: Technical drawing - Transverse support plate (for side camera)

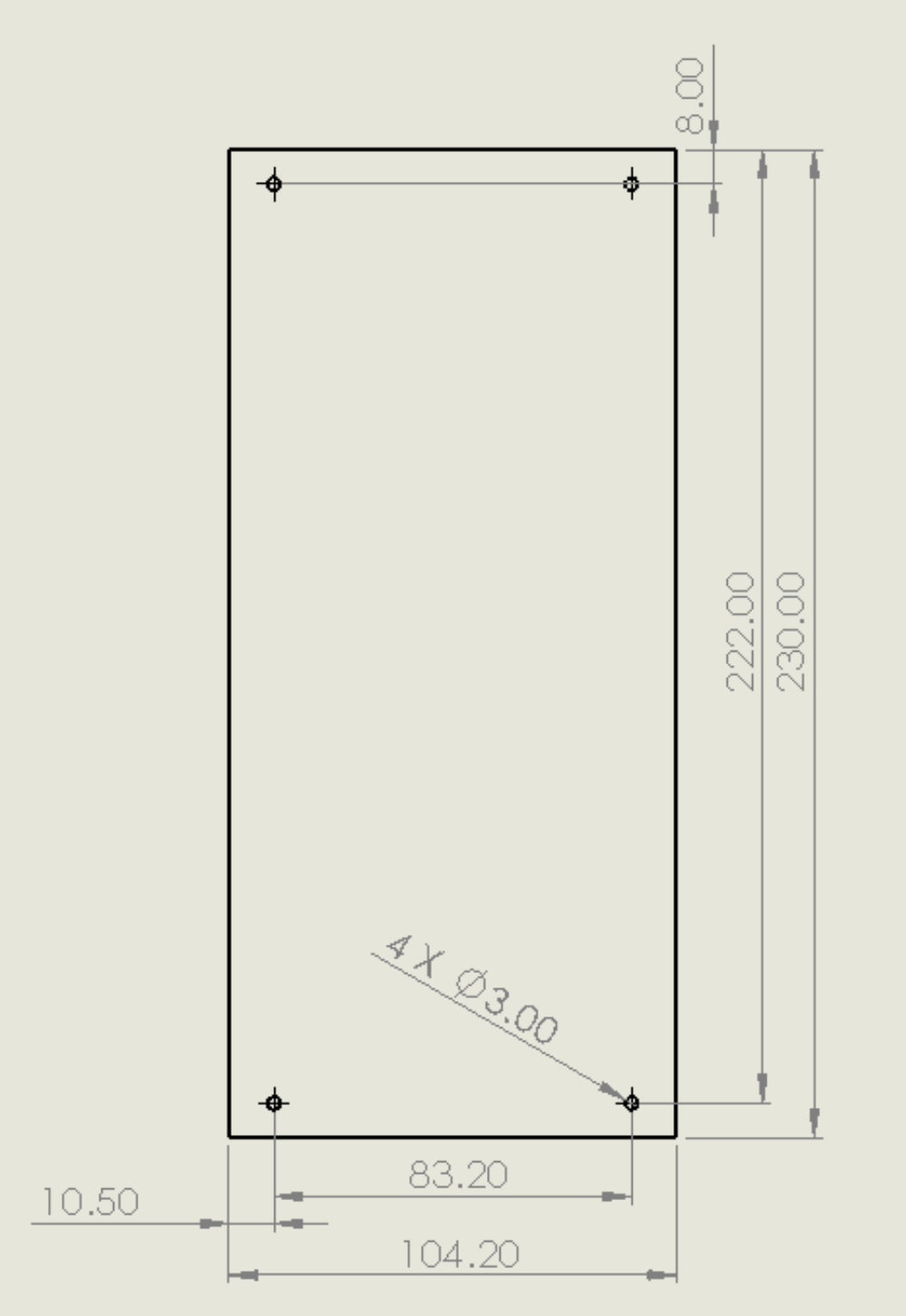


Figure B.2: Technical drawing - Base plate

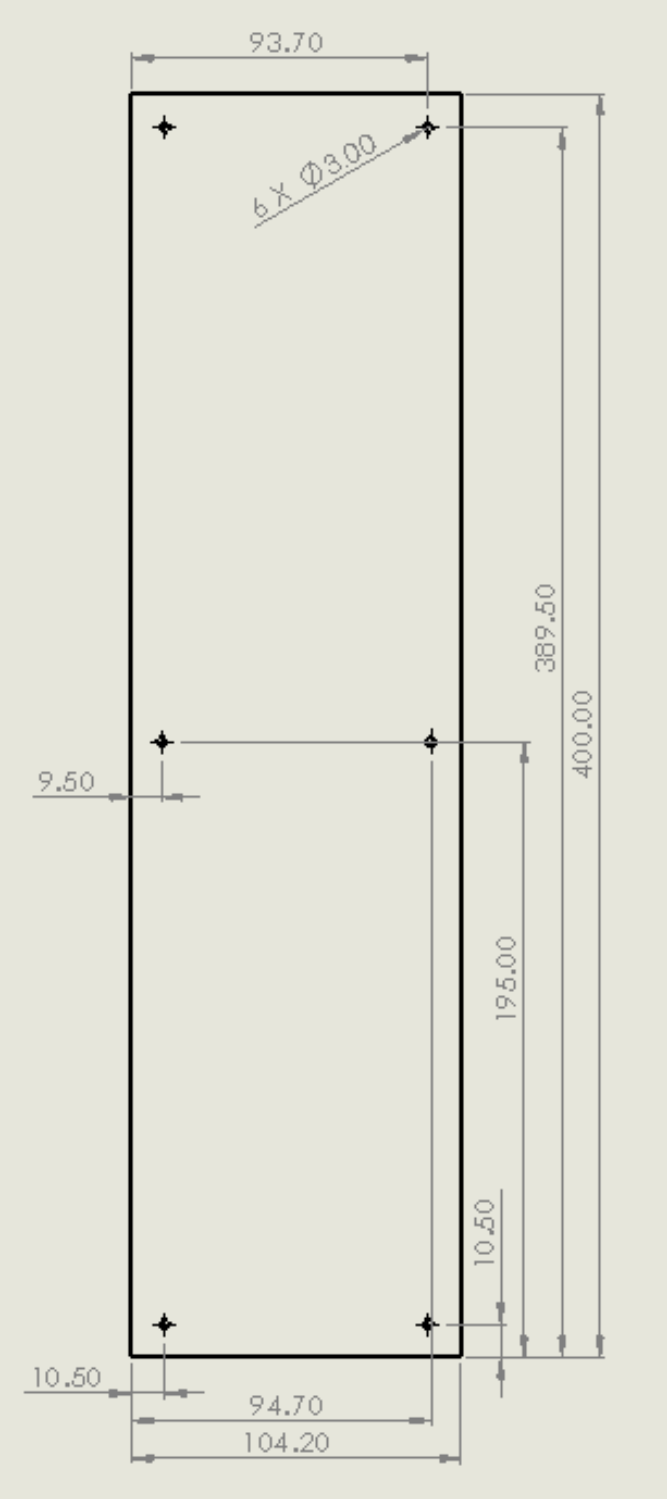


Figure B.3: Technical drawing - Side plate

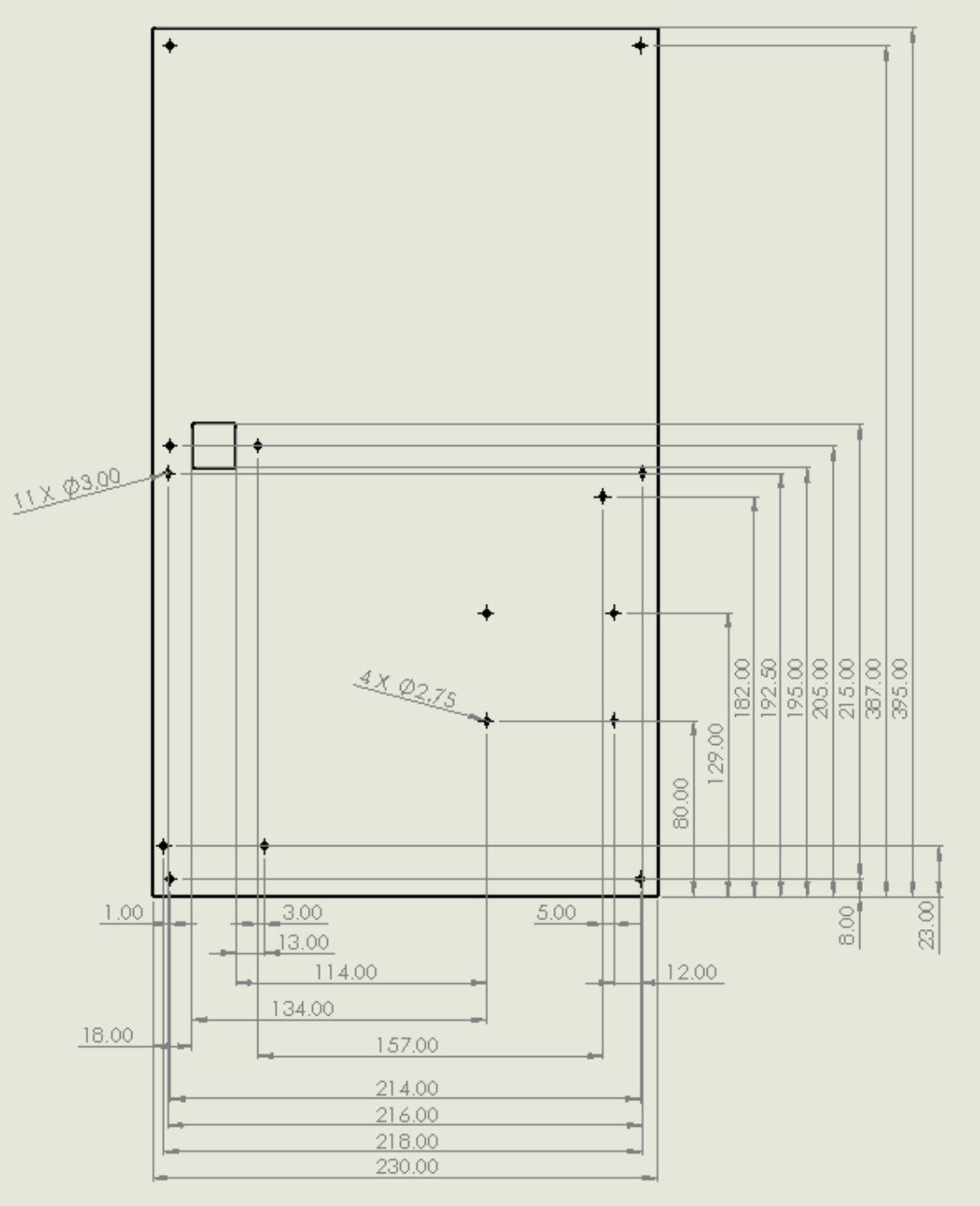


Figure B.4: Technical drawing - Entrance plate

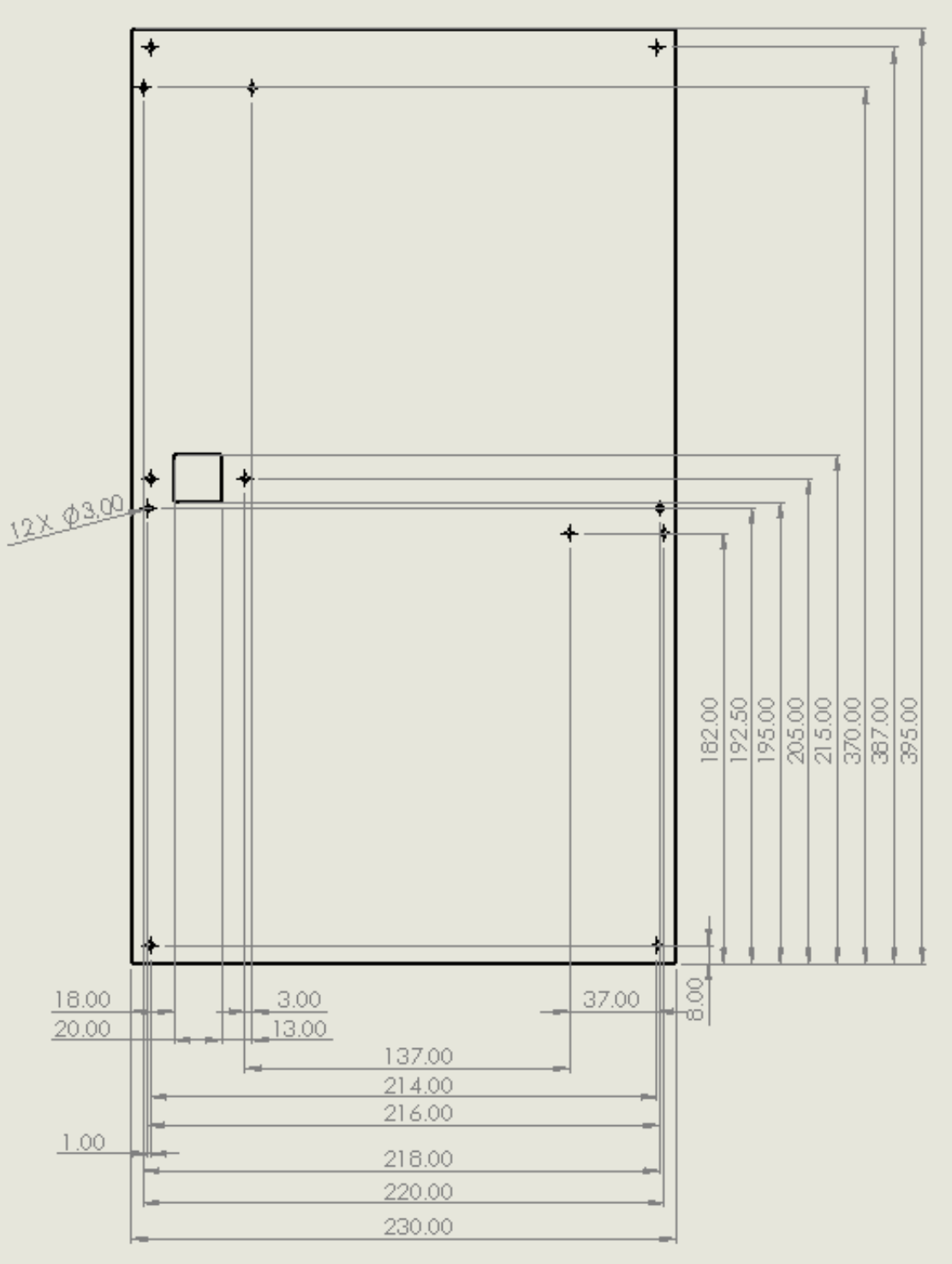


Figure B.5: Technical drawing - exit plate

Appendix C

Python libraries and tricks

C.1 FSWebcam

FSWebcam is an app for image capturing. It allows to save pictures in Portable Network Graphics (PNG) or JPEG formats from different input sources. The command lines are of the form :

```
fswebcam [<options>] <filename> [[<options>] <filename> ... ]
```

In this project, the capture options of interest are :

- `-d` : specifies the device used as input source. In this application, the different cameras can be accessed with `-d\dev\video{NUM}` with `NUM` being the index of the camera.
- `-r` : sets the wanted resolution of the source (`-r640x480`)
- `-s` : sets a control. This allows to modify the camera settings that were in described in subsection 4.3.3. For example, we can change the gain by writing `-s gain = 5`.

In addition, we can specify output options as:

- `--no-banner` : to remove the banner with the details of the image.
- `--jpeg <factor>` : use JPEG as format for the output. The factor can take values from 0 to 95 for the compression.
- `--save <filename>` : sets the name for the file to save.

More information can be found in [40].

C.2 FFMPEG

FFMPEG is a converter for video and audio. It is generally used by specifying input files (for example actual files, device streams ...) and output files with associated parameters. Typical command lines are taking the following form:

```
ffmpeg [global_options] {[input_file_options] -i input_url} ...
{[output_file_options] output_url} ...
```

In this application, the options used are :

- `-i` : specifies the input source. In this application, the different cameras can be accessed with `-i \dev\video{NUM}` with `NUM` being the index of the camera.
- `-s <size>` : sets the wanted resolution for the source (`-s 640x480`).
- `-y` : allows overwriting files without asking.
- `-b:v <bitrate>` : sets the video bitrate to the specified value.
- `-vframes <value>` : sets the number of frames to record to the specified value.
- `-r <framerate>` : sets the framerate to the specified value.

The complete documentation of ffmpeg can be found in [41].

C.3 OpenCV

OpenCV (Open Source Computer Vision Library) is an open source library for computer-vision applications. In this project, the functions that were used to manipulate images are the following:

- `cv.imread(filename,format)`: read an image as an array of pixels from the file specified by `filename`. The second argument `format` allows to select the way to read the image. It can take the following values:
  - `IMREAD_COLOR` : loads the image in the BGR-8 bit color format.
  - `IMREAD_UNCHANGED` : loads the image in its current format.
  - `IMREAD_GRAYSCALE` : loads the image in gray scale intensity format.
- `cv.imwrite(filename,input_array)`: writes the image represented in `input_array` in the file specified by `filename`.

More information about those functions and OpenCV can be found in [42].

C.4 How to run a code at boot

To run the code of the project `Triggered_cam_final.py`, one should type in terminal :

```
sudo nano /etc/rc.local
```

Then add those lines at the end of the script:

```
python3 /home/pi/Desktop/Triggered_cam_final.py &
exit 0
```

Appendix D

Experimental conditions for outdoor trap testing

The following table groups the experimental conditions of the different outdoor tests realized with the insect monitoring device. The weather conditions were extracted from [43].

	Date and hours	**Exact Location**	**Weather conditions**	**Observations**
1	28 April 2022 3 pm to 6 pm	Lat: 41.32211 Lon: -72.92517	Mean Temperature : 12 $[°C]$ Humidity : 22 $[\%]$ Wind : $\approx$ 33 $[km/h]$ Sunny	- No insect detected - Home-made trap (Figure D.1) does not work: - Insects do not fly through - no convergence towards monitoring box
2	12 May 2022 12:30 to 4:30 pm	Lat: 41.32167 Lon: -72.92464	Mean Temperature : 20 $[°C]$ Humidity : 77 $[\%]$ Wind : $\approx$ 15 $[km/h]$ Scattered clouds	- 1 insect observed in the collecting box - IR sensor is fooled by sunlight - Captured images are over-exposed (high ambient illumination)
3	14 May 2022 1:30 pm to 4:30 pm	Lat: 41.32167 Lon: -72.92464	Mean Temperature : 22 $[°C]$ Humidity : 84 $[\%]$ Wind : $\approx$ 9 $[km/h]$ Scattered clouds	- 1 bee before entrance of the trap - Flexible interface piece too narrow - Use a black towel to shield device from sunlight (Figure D.2)
4	18 May 2022 2:30 pm to 5:30 pm	Lat: 41.32167 Lon: -72.92464	Mean Temperature : 20 $[°C]$ Humidity : 37 $[\%]$ Wind : $\approx$ 20 $[km/h]$ Sunny	- No insect detected - Some insects in the bottle before the entrance (Figure D.3) - Use interface piece in Figure 3.24a - Box painted in black
5	19 May 2022 3 pm to 5:30 pm	Lat: 41.32167 Lon: -72.92464	Mean Temperature : 15 $[°C]$ Humidity : 84 $[\%]$ Wind : $\approx$ 11 $[km/h]$ Low clouds	- No insect detected - Some insects in the bottle before the entrance
6	20 May 2022 9:30 am to 5:30 pm	Lat: 41.32147 Lon: -72.92435	Mean Temperature : 17 $[°C]$ Humidity : 84 $[\%]$ Wind : $\approx$ 16 $[km/h]$ Mostly cloudy	- No insect detected - Some insects in the bottle before the entrance - Use of jam and honey as bait
7	21 May 2022 11 am to 5:30 pm	Lat: 41.32147 Lon: -72.92435	Mean Temperature : 23 $[°C]$ Humidity : 79 $[\%]$ Wind : $\approx$ 9 $[km/h]$ Fog (morning) then sunny	- 2 insects observed, more after uninstallation - Lots of insects in the malaise nets (Figure D.5) + bottle before trap entrance (Figure D.4) - Use jam as bait - Lots of insects in the bottle preceding the monitoring device entrance

Figure D.1: Home-made malaise trap

Figure D.2: Trap installed outside with cover on the monitoring box

Figure D.3: Collecting bottle of the malaise trap with few insects

Figure D.4: Collecting bottle of the malaise trap with lots of insects

Figure D.5: Inside of the malaise trap (converging nets)

Bibliography

[1] David Grimaldi and Michael S Engel. *Evolution of the Insects*. Cambridge University Press, 2005.

[2] David L. Wagner. *Insect declines in the anthropocene*. 2020. DOI: 10.1146/annurev-ento-011019-025151.

[3] Caspar A. Hallmann et al. "More than 75 percent decline over 27 years in total flying insect biomass in protected areas". In: *PLoS ONE* 12 (10 2017). ISSN: 19326203. DOI: 10.1371/journal.pone.0185809.

[4] Huajian Liu, Sang Heon Lee, and Javaan Singh Chahl. *A review of recent sensing technologies to detect invertebrates on crops*. 2017. DOI: 10.1007/s11119-016-9473-6.

[5] Todor Ganchev, Ilyas Potamitis, and Nikos Fakotakis. "Acoustic monitoring of singing insects". In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* 4 (2007). ISSN: 15206149. DOI: 10.1109/ICASSP.2007.367014.

[6] Richard Mankin and David W Hagstrum. "Stored Product Protection". In: 2012. Chap. 22 Acoustic Monitoring of Insects.

[7] Ilyas Potamitis, Panagiotis Eliopoulos, and Iraklis Rigakis. "Automated remote insect surveillance at a global scale and the internet of things". In: *Robotics* 6 (3 2017). ISSN: 22186581. DOI: 10.3390/robotics6030019.

[8] Graham A. Montgomery et al. "Standards and Best Practices for Monitoring and Benchmarking Insects". In: *Frontiers in Ecology and Evolution* 8 (2021). ISSN: 2296701X. DOI: 10.3389/fevo.2020.579193.

[9] P Meek, G Ballard, and P J S Fleming. *An introduction to camera trapping for wildlife surveys in Australia | ResearchGate*. 2012.

[10] Roland Kays et al. "Camera traps as sensor networks for monitoring animal communities". In: *IEEE 34th Conference on Local Computer Networks* (2009). ISSN: 2047-0037. DOI: 10.1109/LCN.2009.5355046.

[11] Dustin J. Welbourne et al. “How do passive infrared triggered camera traps operate and why does it matter? Breaking down common misconceptions”. In: *Remote Sensing in Ecology and Conservation* 2 (2 2016). ISSN: 20563485. DOI: 10.1002/rse2.20.

[12] Wikipedia. *Black body*. URL: https://en.wikipedia.org/wiki/Black_body.

[13] C.B.Honsberg and S.G.Bowden. *Photovoltaics Education Website*. 2019. URL: www.pveducation.org.

[14] Jacob Fraden. *Handbook of modern sensors: Physics, designs, and applications, Fifth Edition*. 2016. DOI: 10.1007/978-3-319-19303-8.

[15] B. Heinrich. “Insect thermoregulation”. In: *Endeavour* 19 (1 1995). ISSN: 01609327. DOI: 10.1016/0160-9327(95)98891-I.

[16] Dave Karlsson et al. “The Swedish malaise trap project: A 15 year retrospective on a countrywide insect inventory”. In: *Biodiversity Data Journal* 8 (2018). ISSN: 13142828. DOI: 10.3897/BDJ.8.e47255.

[17] Miklós Dombos et al. “EDAPHOLOG monitoring system: automatic, real-time detection of soil microarthropods”. In: *Methods in Ecology and Evolution* 8 (3 2017). ISSN: 2041210X. DOI: 10.1111/2041-210X.12662.

[18] Esztella Balla et al. “An opto-electronic sensor-ring to detect arthropods of significantly different body sizes”. In: *Sensors (Switzerland)* 20 (4 2020). ISSN: 14248220. DOI: 10.3390/s20040982.

[19] Ilyas Potamitis, Iraklis Rigakis, and Nicolaos Alexandros Tatlas. “Automated surveillance of fruit flies”. In: *Sensors (Switzerland)* 17 (1 2017). ISSN: 14248220. DOI: 10.3390/s17010110.

[20] Ireneusz Ruczyński et al. “Camera transects as a method to monitor high temporal and spatial ephemerality of flying nocturnal insects”. In: *Methods in Ecology and Evolution* 11 (2 2020). ISSN: 2041210X. DOI: 10.1111/2041-210X.13339.

[21] Kim Bjerge et al. “An automated light trap to monitor moths (Lepidoptera) using computer vision-based tracking and deep learning”. In: *Sensors (Switzerland)* 21 (2 2021). ISSN: 14248220. DOI: 10.3390/s21020343.

[22] Toke T. Høye et al. “Deep learning and computer vision will transform entomology”. In: *Proceedings of the National Academy of Sciences of the United States of America* 118 (2 2021). ISSN: 10916490. DOI: 10.1073/PNAS.2002545117.

[23] Charlier Gilles and Hick Simon. “Automated nocturnal insect monitoring in real-time using YOLOv4.” MA thesis. Ecole polytechnique de Louvain, Université catholique de Louvain., 2021.

[24] Beyraghi Vahid and Thibaut Jonathan. “Automated tracking of insects”. MA thesis. Ecole polytechnique de Louvain, Université catholique de Louvain., 2021.

[25] Naturalis Biodiversity Center. *Nature Identification API v1*. 2021. URL: https://identify.biodiversityanalysis.nl/.

[26] Donald A. Neamen. *Semiconductor Physics and Devices Basic Principles*. Vol. 9. 2006.

[27] Electronics Hub. *What is a bypass capacitor*. URL: https://www.electronicshub.org/bypass-capacitor-tutorial/.

[28] *Opto Interrupter ITR20001/T*. DRX-0000278. Rev. 7. Everlight. 2016. URL: https://en.everlight.com/wp-content/plugins/ItemRelationship/product_files/pdf/ITR20001-T.pdf.

[29] *DIY Malaise Net for $69.00*. URL: http://coleoguy.blogspot.com/2010/04/diy-malaise-net-for-6900.html.

[30] Kalitut. *Raspberry Pi USB - Universal Serial Bus*. URL: https://kalitut.com/raspberry-pi-usb-universal-serial-bus/#:~:text=As%5C%20a%5C%20result%5C%2C%5C%20the%5C%20network,connected%5C%20devices%5C%20can%5C%20work%5C%20faster.

[31] David Plowman. *An open source camera stack for Raspberry Pi using libcamera*. URL: https://www.raspberrypi.com/news/an-open-source-camera-stack-for-raspberry-pi-using-libcamera/.

[32] Lalit Maganti. *V4L2 and Hardware Encoding on the Raspberry Pi*. URL: https://lalitm.com/hw-encoding-raspi/.

[33] *Imaging Electronics 101: Basics of Digital Camera Settings for Improved Imaging Results*. URL: https://www.edmundoptics.com/knowledge-center/application-notes/imaging/basics-of-digital-camera-settings-for-improved-imaging-results/.

[34] *Understanding Gamma Correction*. URL: https://www.cambridgeincolour.com/tutorials/gamma-correction.htm.

[35] Stefan Winkler. *Digital Video Quality: Vision Models and Metrics*. 2013. DOI: 10.1002/9780470024065.

[36] Zhou Wang et al. “Image quality assessment: From error visibility to structural similarity”. In: *IEEE Transactions on Image Processing* 13 (4 2004). ISSN: 10577149. DOI: 10.1109/TIP.2003.819861.

[37] Warren Gay. *Advanced Raspberry Pi. Raspbian Linux and GPIO Integration*. 2018. DOI: 10.1007/978-1-4842-3948-3.

[38] *How much storage do I need for Raspberry Pi, Raspberry Tips*. URL: https://raspberrytips.com/how-much-storage-raspberry-pi/.

[39] *Sunlight Inside, Comparing natural light and artificial light*. URL: https://www.sunlightinside.com/light-and-health/natural-light-vs-artificial-light/.

[40] Philip Heron. *fswebcam - Small and simple webcam for *nix*. URL: http://manpages.ubuntu.com/manpages/bionic/man1/fswebcam.1.html#author.

[41] *ffmpeg Documentation*. URL: https://ffmpeg.org/ffmpeg.html#Video-Options.

[42] *OpenCV. Image file reading and writing*. URL: https://docs.opencv.org/4.x/d4/da8/group__imgcodecs.html#ga288b8b3da0892bd651fce07b3bbd3a56.

[43] *April and May 2022 Weather in New Haven, Connecticut, USA*. URL: https://www.timeanddate.com/weather/usa/new-haven/historic?month=5&year=2022.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**