IT & Robotic Labs

Project: Control Table

Robotics & Automation pathway

Authors	: Arnold Odhiambo, Charles Ebelle, Bernard Szeliga, Eugenie N'Diaye, Elijah Ki-Zerbo
Scrumaster	: Bernard Szeliga

Version	: 1 R 0
Promotion	: EENG4
Cursus	: EENG
Date	: 08/01/2023

Versions

Version	Date	Author	*Kind of modifications*
1 R 0	01/10/2022	Elijah Ki-Zerbo	**Creation**
2 R 0	16/10/2022	Elijah Ki-Zerbo	**Update after lab on the 11th**
3 R 0	28/10/2022	Bernard Szeliga	**Update after lab on the 25th**
4 R 0	27/11/2022	Bernard Szeliga	**Update after lab on the 21st**
5 R 0	10/12/2022	Bernard Szeliga	**Update after lab on 7th**
6 R 0	08/01/2023	Bernard Szeliga	**Final report**

A.Table of Contents

B. Table of Figures

C. Introduction

As students of the robotic pathway at ECAM we are supposed, based on the experience of our applied internship, to complete missions and to apply the skills we understood the last year to actual projects of robotics and/or informatics.

In the IT and Robotics courses, a new mandatory course that the Robotic pathway students have, we were given the task of carrying out a project. It was to be realized within 7 sessions interval while making use of the scrum methodology introduced to us at the beginning of the project. This is a continuation of last year's program.

Scrum is a framework for project management for teams between 3 to 9 developer and 1 Scrum master relying on breaking the work into goals that can be done in sprints (periods when a scrum team works to complete a set amount of work) after carefully planning the day's objectives, after what there will be a review session and a retrospective on the work actually completed.

This opposes the usual V-model minimizing risks but only producing value after a long time. We will be applying this new methodology to the project we chose: a ball-controlling table using fans.

This report will initially describe the project and the tool involved in its realization, then there is a presentation of the different methods utilized followed by the results issued by each of these methods to finally go over the management of the team and give a conclusion on the session.

D.Project description

The project was to be made within groups of 5. The main reason that this project was carried out in groups is that it gives us the chance to practice skills necessary for processing information, analyzing problems and solving them, as well as management skills and evaluation skills necessary for coming up with solutions.

The main goal of the project is to create a system that can control the position of a ball regardless of its initial disturbances or position (Figure 1)

Figure 1. Provided image of the setup

The ball could move all over the board, thus, the goal is to ensure that at any given position of the ball, we are able to detect it position and move it to the center of the table. We then adjust the speed of fans (Error! Reference source not found.) depending on the position to control the ball. Finally, we should potentially be able to have the ball follow more complex path (Figure 2)

Figure 2 Provided image of a more complex setup

E. Methods

1. Equipment

1.1. Camera

A Logitech camera whose purpose of the camera is to detect the ball's position so the fans can be activated accordingly (Figure 3). The camera, which is a Logitech c615, has the following specifications [1]:

- Max Resolution: 1080p/30 fps-720p/ 30fps
- Focus type: Autofocus
- Lens technology: Glass
- Built-in mic: Mono
- FoV: 78°

Figure 3. Picture of the used camera

1.2. Fans

The fans are vital in this project as they do the actual control of the ball. By increasing or decreasing the speed of the fan, or turning it off completely, the ball can be controlled.

For the purposes of this project Foxconn 432768-001 rev.d (Figure 4) are used with nominal voltage of 12V, peak current of 0.4A and PWM control.

Figure 4. Picture of used Foxconn 432768-001 rev.d fan

1.3. Power supply

Generally, the primary function of the power supply is to convert the electric current from the source into the necessary voltage, frequency, and current to power the load. Additional power supply (Figure 5) is required for the project because power rating of Arduino is significantly to low as well as maximal provided voltage is equal to 5V while nominal voltage of the fans is equal to 12V.

Provided power supply is a Delta Electronics PGS-400ab ATX power supply with important parameters from project perspective [2]:

- Maximum power: 400W
- Maximum load:
    - +12V1 rail: 14A
    - +12V2 rail: 16A

As only 12V rails will be used, and fans are PWM controlled as opposed to voltage control, maximum load on other rails as well as voltage ripple does not impact the operation of the system.

Figure 5. Power supply used for the project

2. Programming languages and libraries

2.1. C++

For desktop application, C++ programming language was used. It was chosen due to its speed, compatibility with libraries, as well as for familiarity of the team with it that was required for project with relatively short schedule.

2.2. C Arduino

For used microcontroller Arduino version of C programming language was used. For this family of microcontrollers and embedded computers it is only reasonable option to use. Potential alternative would be to use embedded python, but interpreter overhead could result in significant performance issues.

2.3. OpenCV

OpenCV is a library of programming functions mainly aimed at real-time computer vision. It was used for operating the camera as well as image processing.

3. Project development

Project can be roughly divided into three distinct parts (Figure 6):

- Hardware
- Arduino software
- PC application

Each part performing its own set of functionalities

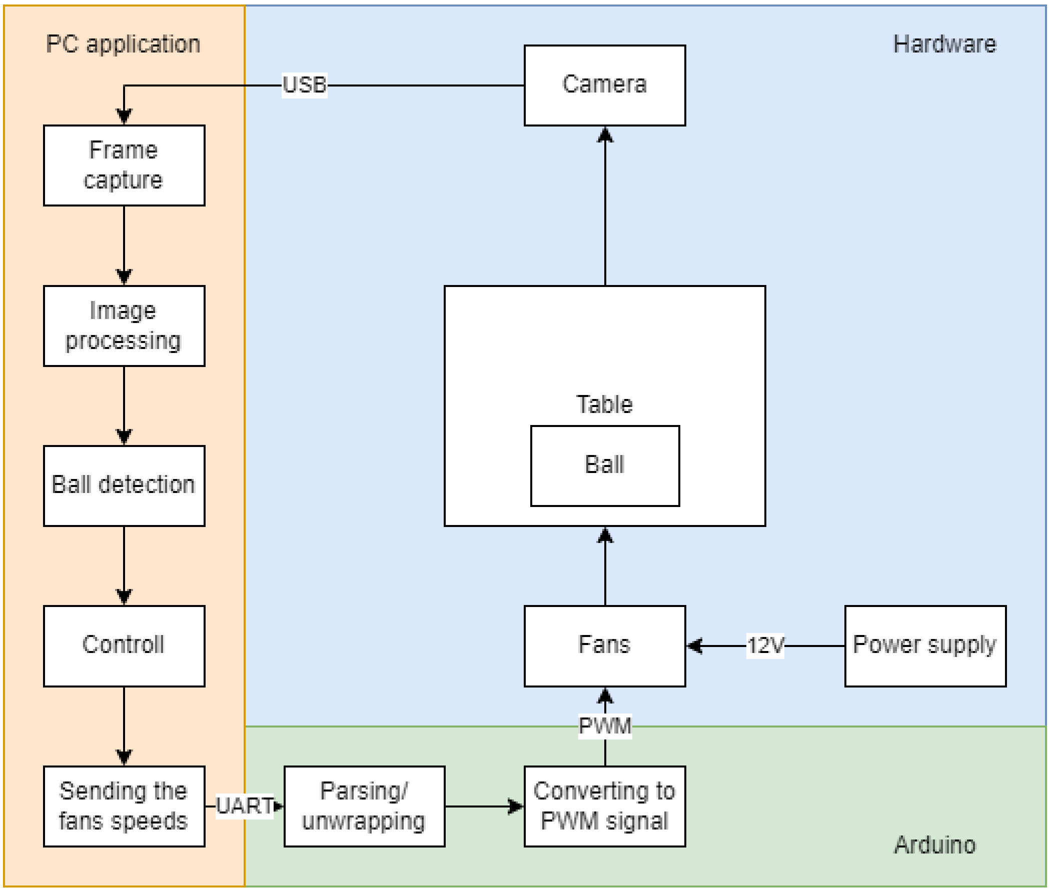


Figure 6. Block diagram of the project

3.1. Hardware

3.1.1. Table

We made a quick setup to have a general idea of how the project would look like (Figure 7). This also helped us know the extra equipment we would possibly need and different things to take into consideration

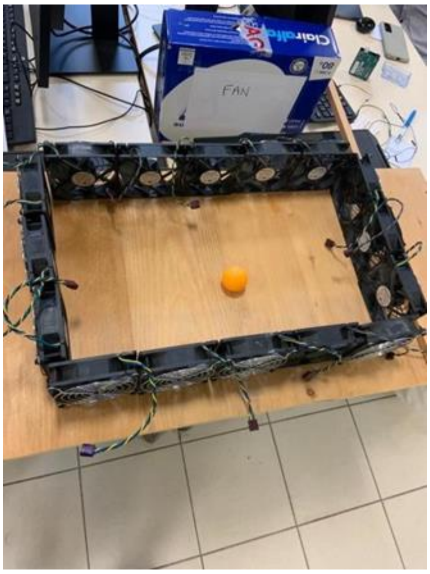


Figure 7. Test fit of the table setup

3.1.2. Fan brackets

For mounting of the fans, brackets were designed (Figure 8a) using standard 92mm fan spacing (Figure 8b). Those brackets provided stability to the fans as well as allowed for maintaining right angles between sides of the table that was required for effective control of the ball.

Created brackets can be connected in order to hold used fans together (Figure 9).

a) b)

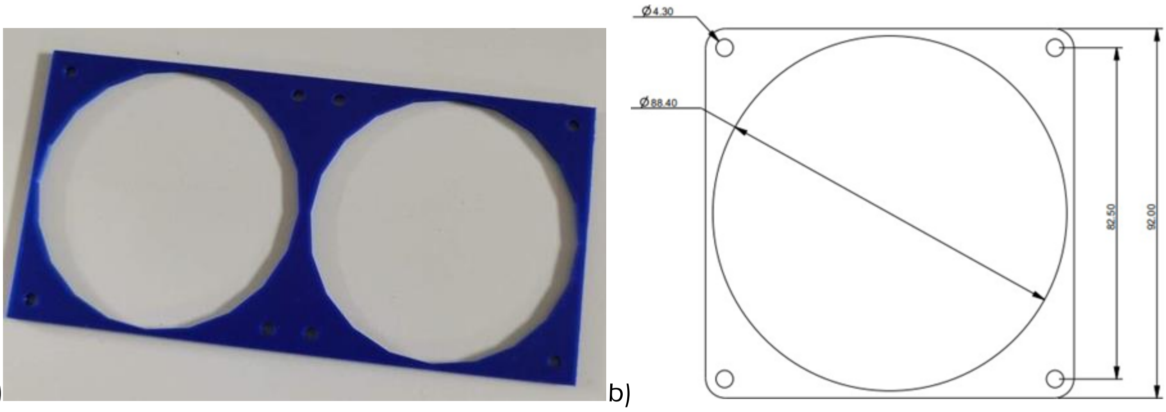


Figure 8. a) Picture of the designed and 3D printed bracket b) Technical drawing of standard 92mm fan spacing

Figure 9. Fans connected with designed brackets

3.1.3. Fan wiring

To power the fans, a separate 12V (and ground) rail had to be used, during testing fans were directly connected to the supply, but for the final product proper rail had to be made as current draw of 16 fans would significantly exceed the rating of the wires (typically 2A). PWM wires will be connected directly to the shield.

3.2. Arduino

3.2.1. Parsing

Parsing of an image generally refers to the decomposition of an image into its constituent visual patterns. It was decided that data would be sent to the processing Arduino board within and array of the following template:

[255, 0, 62, 0, 0, 27, 0, 0, 0, 0, 0]

With each value representing a different fan whose intensity has been mapped on 0 to 255 scale

The Arduino board will then have to have a function capable of interpreting this data individually before adapting each servos output

To do so the strtok and atoi() function would probably be required.

3.2.2. Fan control

For the control of the fan PWM signal is generated by Arduino shield. Communication with the shield is performed via I2C protocol using shields library [3].

As the development of the Fan control script progressed on Arduino, problems arose regarding the fan's sensitivity: While mapping the max speed of the fan to 255 and at rest 0, what would it mean to add 1 to the fan speed value.

To get those answers further trials were required this time using the tachometer output coming from the fan to make a graph of the fan speed evolution against the augmentation of the value contained in the byte (see Parsing)

3.3. PC application

3.3.1. Ball detection

Ball detection requires number of steps (Figure 10) to be taken before position can be calculated

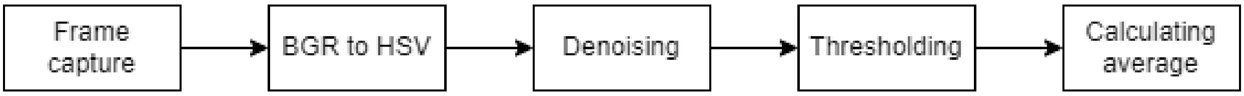


Figure 10. Block diagram of image processing

3.3.1.1. Frame capture

Frame capture is done using OpenCV, a pipeline to the camera is created during initialization and in every cycle of the program one frame is grabbed from the camera for further processing and calculations.

If all actions in one cycle are performed faster than framerate of the camera, the same frame will be processed. While it's not optimal solution from perspective of energy efficiency, it guarantees that always the newest frame will be processed even if cycle rate will drop below framerate of the camera.

3.3.1.2. BGR to HSV conversion

OpenCV captures frames in BGR color space, while for a lot of applications it is not problematic, for color-based ball detection, which is used in the project, HSV color space is more suitable. Using HSV, instead of specifying color intensity ranges for each of the basic colors (red, green, and blue), hue, saturation, and value are specified. This difference is important considering how colors react to different lighting conditions, in BGR color space to ensure that the ball would be "in range" in different conditions ranges of all three values have to be wide as all three are impacted by the light. In HSV color space only value (V) is affected noticeably by the brightness of the light while saturation (S), and hue (H) are almost constant (impacted more by white balance than brightness). As a result, using HSV color space allows for narrower ranges and thus reduces the risk of false-positive while maintaining a low rate of false-negatives (Figure 11).

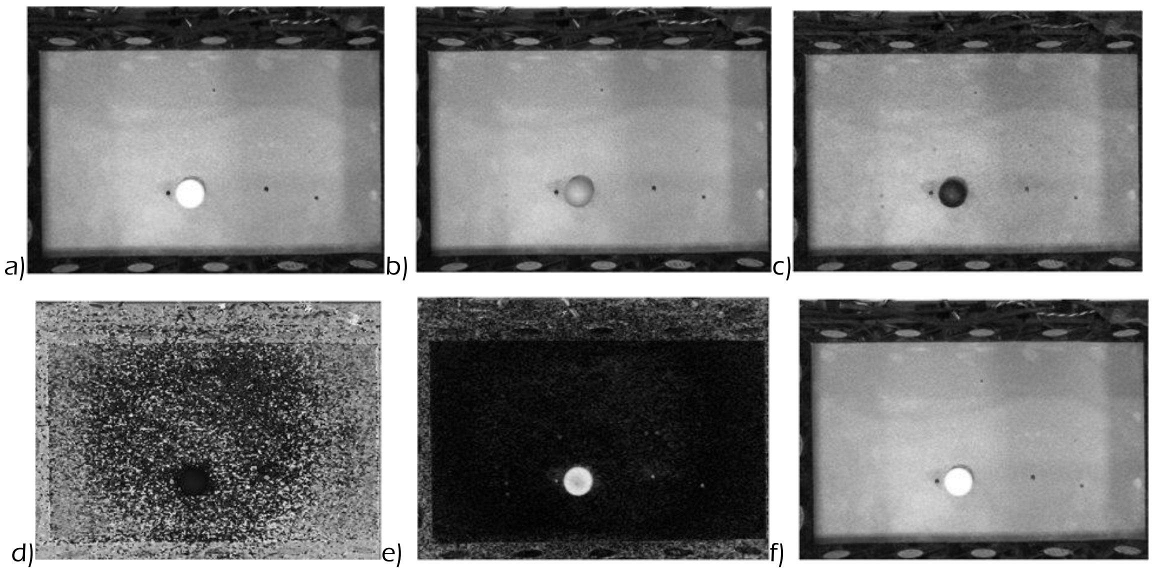
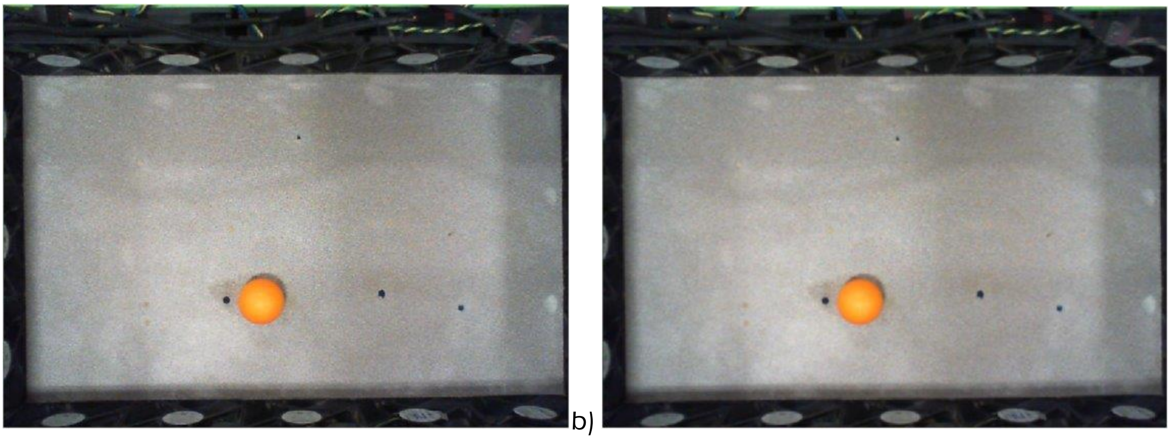


Figure 11. The image separated to Red (a), Green (b), and Blue (c) layers vs separated to Hue (d), Saturation (e), and Value (f) layers

3.3.1.3. Denoising

Most of the web cameras, including one used in the project, use small and relatively cheap sensors that are prone to impulse noise. The problem becomes more pronounced at low light levels when to correct exposure the ISO of the camera needs to be increased resulting in the appearance of "grain". With the location being determined by the average position of thresholded pixels, outliers created by such noise could impact the calculated location of the ball.

To avoid this kind of problem gaussian blur is used which eliminates single pixels of different colors while not impacting the pixels of the ball in a noticeable way (Figure 12).



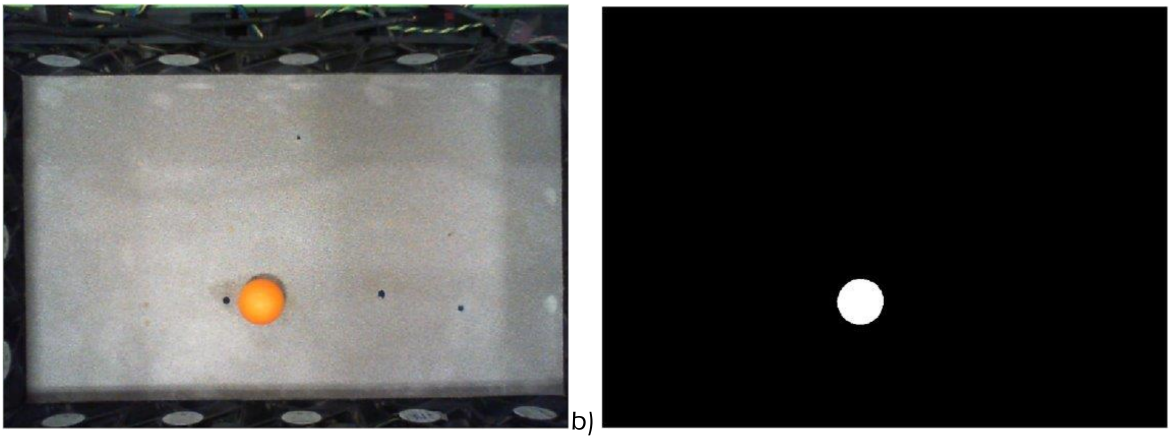
a) b)

Figure 12. The captured frame before (a) and after applying blur (b)

Thanks to the usage of HSV color space and because data from one frame does not impact the next one the automatic white balance, which can have a noticeable impact on color in the first few frames, does not impact the calculated location and thus other calculations.

3.3.1.4. Thresholding

Thresholding sets pixels inside the set range to white and ones outside to black. This conversion creates a clear divide between what is considered to be an object of interest and what is not (Figure 13).



a) b)

Figure 13. The captured frame (a) and created mask (b)

3.3.1.5. Calculating an average

The average position of all pixels that met a condition of the threshold is calculated. For round objects such as balls, the centroid is a good enough approximation and calculation of the outline is not necessary.

3.3.2. Calculating the fan speeds

The control of the fans' speeds can be divided into two parts: selecting "active" fans and calculating speeds. For calculating the speeds of fans two functions were developed (PID and simple) that can be selected by changing definitions in the configuration file.

3.3.2.1. Calculating active fans

The active fans are fans where the calculated position of the ball lies inside their direct airflow cylinder. Active fans are calculated by dividing the workspace into 15 zones (5 by 3 rectangle, due to the number of fans on each side) and selecting fans corresponding to the given zone.

While all fans could be controlled at the same time, this method of control would yield only minor changes to the system's overall operation and would significantly increase the complexity of calculations and thus its time due to the necessity of evaluating interactions between all fans speeds on airflow inside the work area.

3.3.2.2. Simple control

The simple control method calculates speed by inverse proportional distance to the given fan. That is, fan speed increases if the ball is closer to it.

This method can be considered to operate as a P controller and allows only for placing the ball in the center of the work area but it automatically adjusts its K_p value depending on the size of the work area and thus does not need to be calibrated in case of change in camera or its mounting method.

3.3.2.3. PID

The PID controller allows for more sophisticated control allowing theoretically placing the ball at any point in the work area. As a drawback, it requires calibration of K_p, K_i, and K_d values, as well as loop time synchronization for proper operation of integrating and differentiating parts.

To eliminate "kick" in fan speeds when new set-point is selected, differentiating on measurement instead of on error was implemented.

3.3.3. Serial communication

For serial communication third-party, the open-software library is used [4]. The calculated speed of each fan will be transmitted from the PC in form of 12 comma-separated values for ease of parsing on the side of Arduino. Values are normalized in the range of 0 to 255.

With the used range, the speed of each fan could be calculated by one char (8-bit variable) but this approach poses a risk of unintended command characters being sent such as newline characters. To avoid that values can be sent as either base 10 numeral or base 16 numeral, the first option gives better readability for users unfamiliar with using hexadecimal values while the second option would reduce the length of the number by 1 character for each fan resulting in the decrease of 25% of total frame length (when comas and end line are included)

Further tests are required to decide which method will be more beneficial. If communication between PC and Arduino will appear to be a bottleneck, hexadecimal values and potential removal of commas could be used to alleviate overhead at cost of only human readability of code and values (what can be generally considered to be "free")

Due to need for loop time synchronization, communication time had to be shorter than remaining time from loop time after all calculations were completed. During testing calculation time for previous stages combined was equal to 30ms allowing for synchronizing loop rate with framerate of the camera if communication would take less than 3ms. As such Baud rate of 115200 was selected as for this rate, communication takes 0.5ms.

F. Results

1.1. Ball detection

Created ball detection algorithm was successful at detecting ball. With the initial configuration, it was prone to error due to too wide a range of accepted colors. As such, more objects were meeting threshold criteria, resulting in false positives and then incorrect position estimation. After creating more discriminant conditions false positives were significantly reduced and the ball was reliably tracked.

1.2. Calculating the speeds

Calculating the speeds worked as intended, in case of both controlling methods when ball was properly detected during run-time of the program. If ball at any point was not detected it created extreme values at the integrating part of the PID controller as no detection was treated as the minimum value of integer (-2,147,483,648) resulting in constant saturation of the output value.

1.3. Controlling one fan

As already stated above, we began by controlling one fan.   After doing the connection, we ran the code and the fan started moving at a constant speed. We noticed that if we remove the PWM, the fan moves at a significantly higher speed. This is expected as the role of the PWM is to lower an electrical signal's average power by effectively breaking it up into several sections. This was possible at the beginning, but we couldn't notice as the delay was really small. We increased the delay and in the end, we were able to increase and decrease the speed of one fan significantly.

In addition, we were able to control four fans at the same time when they were not at low speed. This is because some fans were not moving at all at a low speed. We thus decided to test all 16 fans individually to see if they were faulty. We were able to test 7 fans and found out that 4 of them were faulty. We highlighted them with a marker and the professor was made aware of this as well. In the next lab, we plan to finish testing the rest of the fans.

Once again, the group retested the fans and all of them seemed to be working perfectly at all speeds. This was surprising as even the ones that had initially been labeled as faulty started working. This could possibly have been caused by wrong wiring at the beginning when testing the fans, however, it was now confirmed that they all work perfectly.

1.4. Parsing of the Image processing data

However, as this was being done, a light problem came up as the data would be sent under the format of a string, a collection of letters, making it impossible for the Arduino to decipher it. Therefore, we had to implement a function within our board to transcript each cell of the array to an integer value.

The atoi() function, a pre-implemented Arduino function that converts strings into integers, was used to develop a parser. This parser scans an array, detecting each comma and recognizing them as the end of the first integer. It then uses the atoi function to copy the integer into a pre-made integer array.

There was however a problem encountered when parsing the date due to the way the buffer of data received from the PC was implemented. This caused the compiler to misinterpret the value being sent to the Arduino, resulting in incorrect operation of the fan.

An attempt was made to solve this issue by splitting the data frame and sending the speed to the microcontroller one by one, but this did not resolve the issue. It was discovered, however, that inputting the data directly from the Serial Monitor resulted in correct system response, indicating that more time spent on resolving this issue would result in functional system operation.

1.5. Mounting the Equipment

After properly putting the mounting brackets in place, the group decided to put all the equipment of the setup together. This was to enable us to see how the setup would look and to know if the height of the camera that had been initially estimated would be enough. At the beginning, the group had estimated that the camera would be placed at a height of approximately 40cm. After putting the equipment together and seeing the setup, it was decided that a height of 50 cm would be more appropriate. Due to the addition of the mounting brackets, the fans were able to fit well together and seem to be more stable.

G.Team management

We used a method known as scrum to carry out this project. Scrum is a technique commonly used, especially in the development of software. It is focused on incremental and iterative procedures known as sprints. Sprints basically refer to a brief window of time within which a scrum team is required to finish a specific amount of work.

This opposes the usual V-model minimizing risks but only producing value after a long time. We will be applying this new methodology to the project we chose: a ball-controlling table using fans.

We wrote down the list of all the tasks we would have to do during this project. We also gave an approximate time of how long it would take us to finish each individual task. This was done on Wekan by attributing points (Table 1). These points were Fibonacci numbers and they corresponded between the sprint point and the duration of a task, with two being the minimum and twenty-one being the maximum. In addition, we were able to appoint a scrum master. The scrum master improves and streamlines the team's methods for achieving its objectives.

Table 1. Relation between the number of sprint points and the time estimated for a task

Number of sprint points	Corresponding time
1	10min
2	20min
3	30min
5	1h
8	2h
13	3h
21	4h

Tasks are divided into several categories (Figure 14). The backlog is a list of work items the team plans to complete during a project sprint. A task is then moved to the 'to do' section just before it's done and then to the 'in progress' section as it is being done. After the task is completed, you move it to the 'done' section then later when the team is ready to test it, it is moved to the 'to test' section.

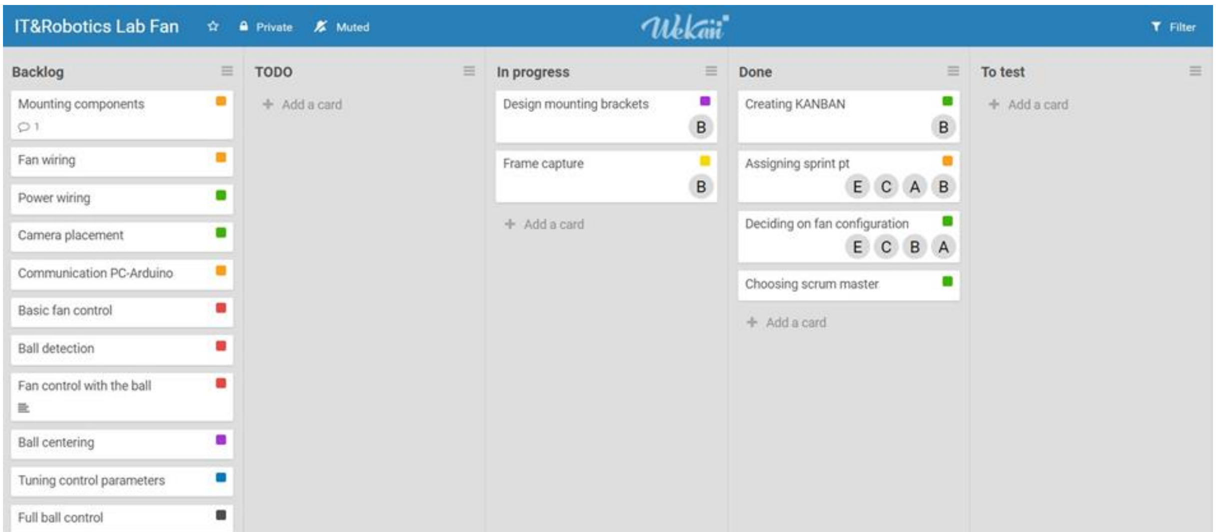


Figure 14. Kanban board created in Wekan

This scrum technique that was introduced to us at the beginning of the project by the professor significantly shortens the time to benefits and increases productivity.

Control table report

We were also able to divide tasks for us to work more efficiently and quicker. This also helps members of the group to work on their strengths and improve on their weaknesses.

Due to the absence of the scrum master due to health reasons at the laboratory meeting on 11.10, sprint progress was not correctly recorded.

During the initial sprint meeting during the 25.10 session 10 points were targeted (5 for ball detection and 5 for fan control) but due to higher than expected rate of work this target was expanded during the mid-sprint meeting by adding serial communication (3 points)

It can be observed that as time progresses the number of finished points is approaching the target velocity (Figure 15), but the initial errors in the estimation of the time taken by all actions carry its impact, so the points performed so far are closer to the pessimistic scenario (Figure 16). It also can be seen, that the estimation error for each week oscillates but approaches 0 (Figure 17).

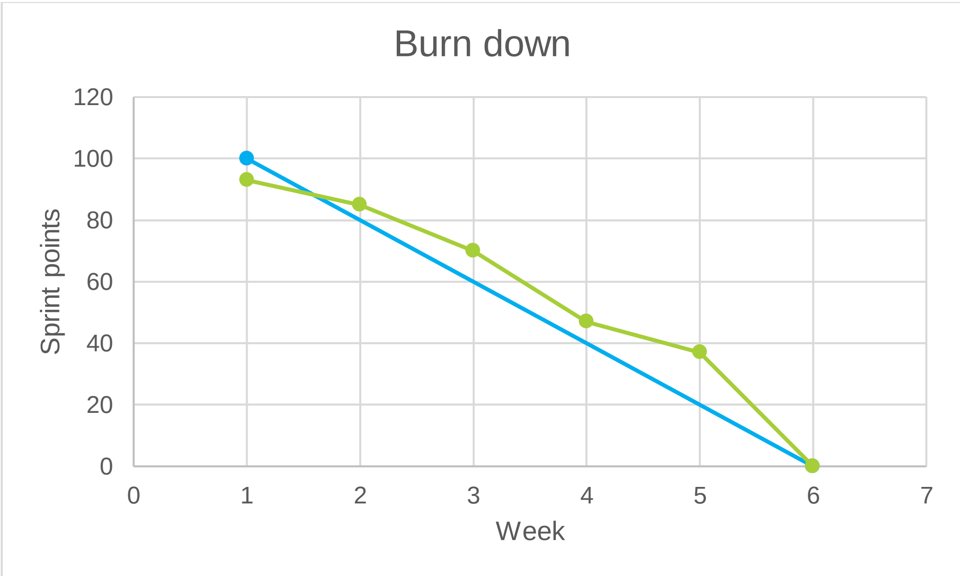


Figure 15. Burn-down chart

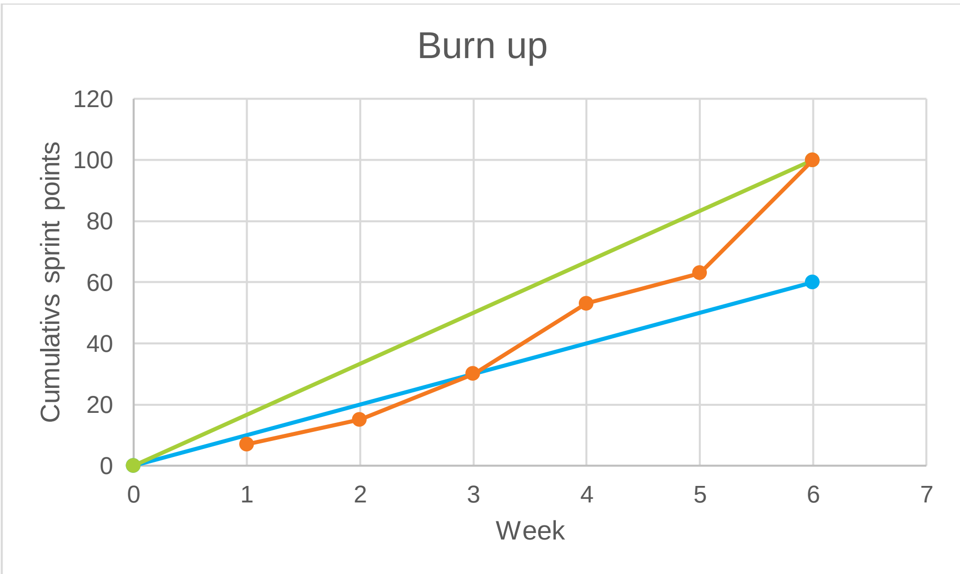


Figure 16. Burn-up chart

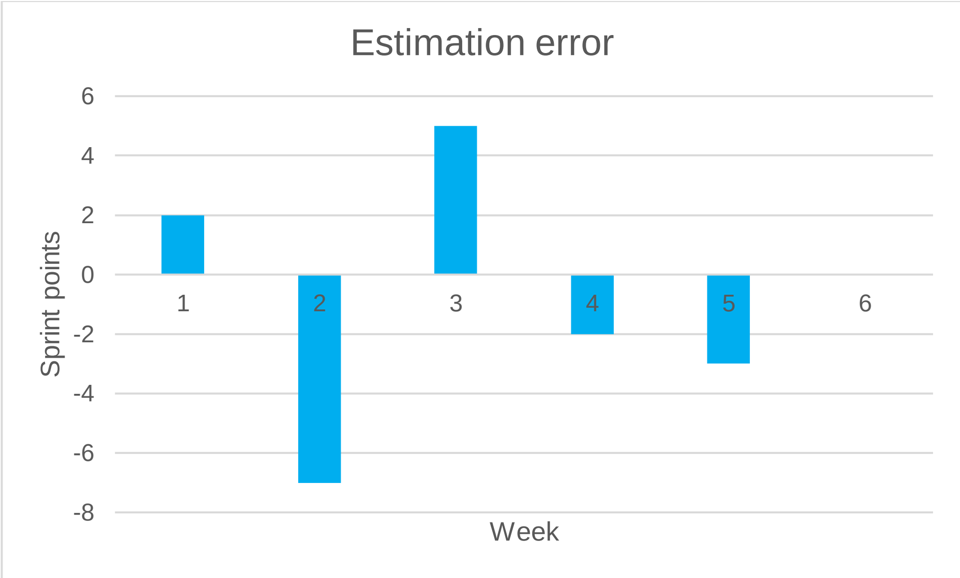


Figure 17. Chart of estimation error over the weeks

H.Discussion

During the development of the projects some additional problems has been discovered and need to be addressed:

- If the camera was not properly mounted (directly above the center of the table) the parallax error could be observed. To ensure proper operation of the table even in those situations more complex control system could be implemented for active fan selection that would define work area using 4 points instead of 2 but his change would increase calculation time and thus prevent synchronization with framerate.
- Created breadboard used for connection of all fans had a tendency to disconnect fans when moved due to mounting pins being designed for PCB and thus being to short for mounting on bread board. To prevent that and also ensure proper thickness of power connection, dedicated PCB could be created.
- Lack of prior knowledge on the project or its requirements resulted in created task list being incomplete and thus, even though according to created charts project is full completed, not all functionalities are working as intended.
- While deciding values of the sprint points for each tasks, values were decided in function of teams time instead of single member and thus total value for each sprint and total project should be multiplied by 5 to properly corelate to worktime of the project

I. Conclusion & Remarks

In conclusion, the subject we were given was a control table. This should allow us to control the position of a ball placed inside it. In other words, we must be able to maintain the ball in a given position despite the degree of inclination that the table could have or move the ball to a given position without having to touch it directly. During this session we determined the elements necessary for the realization of our project and defined the tasks to be fulfilled on the short and long term of our project using the SCRUM method.

Also, we were able to find out some small details we had not thought about could be critical for this project. For instance, we had not thought about testing the fans to check for possibilities of a fault.

In addition, we are able to use our previous knowledge acquired during lessons such as signal processing and coding, to come up with solutions for this project. This helps us appreciate the knowledge more and understand it even better. Also, this project has also helped us identify each other's strengths and weaknesses, which makes it easier to work together.

By the end of the project group was able to finish all task set at the beginning of the project, but number of additional task had to be completed for full operation of the system.

J. Bibliography

[1] Logitech, «Logitech c614 product page,» [En ligne]. Available: https://www.logitech.com/en-us/products/webcams/c615-webcam.960-000733.html. [Accès le 21 11 2022].

[2] Delta Electronics, *GPS-400AB datasheet.*

[3] Adafruit, «Adafruit-PWM-Servo-Driver-Library,» 29 6 2022. [En ligne]. Available: https://github.com/adafruit/Adafruit-PWM-Servo-Driver-Library. [Accès le 7 12 2022].

[4] dmicha16, «simple_serial_port,» 28 8 2021. [En ligne]. Available: https://github.com/dmicha16/simple_serial_port.git. [Accès le 25 10 2022].

K.Appendix A

Parsing function

```
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <limits.h>
#include <sys/types.h>

int* parser(char* x){

  char line[100];
  strcpy(line,x);

  char *separator = ",";
  char str[16][3];
      int i = 0;

      char *token = strtok(line, separator);
      do
      {
            strcpy(str[i++], token);
      }while ((i < 16) && ((token = strtok(NULL, separator)) != NULL));

    int y = 0;
    int speed[16];
    char val[3];
    for(y=0; y<=15; y++){
        memcpy(val, str+y, 3);
        speed[y]=atoi(val);
  }
```