

Smart head-mounted LED lights

Table des matières

1	Project description	1
2	System overview	2
3	LED lighting module	2
3.1	the regional distribution of light intensity on the target surface	2
3.2	Control the intensity of LED lights	6
4	Photodiode detection module	6
4.1	The physical composition of the detection system	6
4.2	Low-pass filter system	8
5	Feedback correction module	9
6	Automatic distance and angle measurement module	11
6.1	Automatic distance measurement	11
6.2	Automatic angle measurement	12
7	Data transmission module	15
8	Interface graphique utilisateur	16
8.1	Choix de l'application	16
8.2	Recherche du périphérique bluetooth	16
8.3	Sélection du monde et envoi d'information	17
8.4	Installation du code sur smartphone	17
9	Design du Circuit	18
10	Conclusion and amélioration	19

1 Project description

Our goal is to achieve uniform illumination of the target plane by adjusting the LED light array, even if there are random effects from external light sources. We draw a schema as figure 1.1 to help people better understand our project.

The product can be used in a variety of scenes, such as when people work at the desk, whether they turn on the table lamp or pull the curtains, the smart LED lights can produce uniform lighting on the desk, or when people run outdoors at night, we can get uniform lighting on the ground. It can also be used by a person who wants to read a book in a dark place.

In order to deal with these complex situations, we need to add an automatic measurement

function to the product, so that it can automatically measure the current height and the angle of tilt according to the change of the user's head position. Then according to the measured data, the luminous intensity of each LED is adjusted in real time to maintain uniform illumination of the target surface.

On the other hand, we also need to design a software on the smartphone in order to allow the user to send their expected luminance of the target surface into the product.

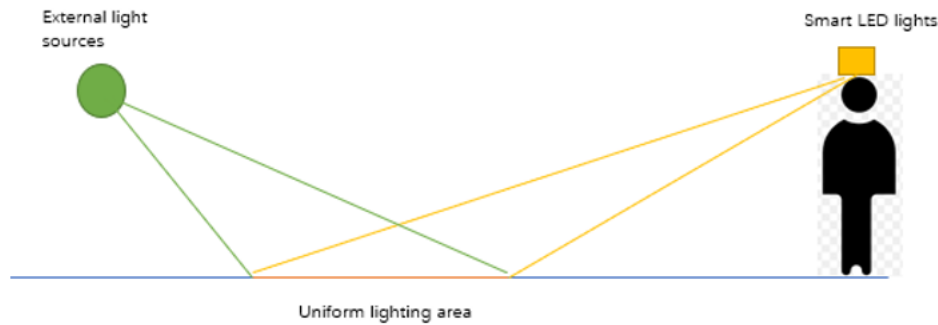


FIGURE 1: Project application concept schema

2 System overview

Before introducing the technology used by each module in detail, we would like to introduce the system as a whole.

The core part of the system is a 3×3 LED array and a 3×3 photodiode array. First, we need to use the mathematical relationship to calculate the area that can be illuminated on the target surface. Here we need to use the automatically measured height and angle data. Then we use the 3×3 photodiode array to detect the object region and filter, here we take into account both the light emitted by the LED array and the light emitted by the external light source. Next, we use the data obtained by the photodiode detection to compare with the expected value input by the smartphone, and then calculate the luminous intensity that needs to be modified for each LED, in this step, we will use the feedback system learned in automation to achieve it.

In the following section, we will introduce in detail how to implement the system in sub-modules. It should be noted that the system we built has a wide range of adaptability. Manufacturers can modify the system according to the methods mentioned in the following technical report, such as changing the opening angle of the LED array, changing the type of LED, changing the type of detector, etc. For users, our system is very convenient, just turn on the phone, enter the desired luminance value, and other parts of the system will be realized automatically.

3 LED lighting module

In this part, we will introduce the technology used in the LED lighting module, which include the calculation of lighting area and the control of LED's intensity.

3.1 the regional distribution of light intensity on the target surface

The first thing we need to do is calculating the area where can be lighted by our LED array. We need to define some important parameters which are necessary when we do the calculation,

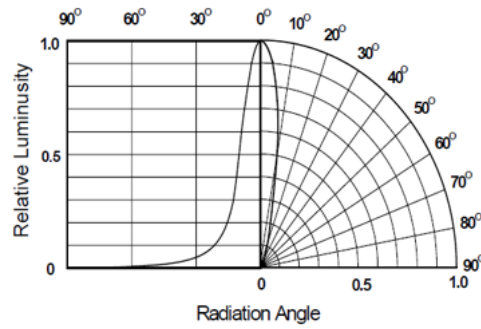


FIGURE 2: the radiation angle distribution of LED 334-15/T1C1-4WYA

in the figure ??, for each pixel, he has two opening angles of horizontal direction ($\beta_{internal}$) and vertical direction ($\alpha_{internal}$), we can define them according to the datasheet of LED.

In our simulation, we choose a type of white LED, 334-15/T1C1-4WYA of everlight company, whose radiation angle distribution is shown at figure 3 According to this figure, we can consider that the opening angles $\alpha_{internal}$ and $\beta_{internal}$ are 7° , the relative luminosity is 0.8 at this angle. The system is symmetrical. If the manufacturer wants to use other type of LEDs, he only needs to modify the corresponding $\alpha_{internal}$ and $\beta_{internal}$ values in the matlab main program according to the radiation angle diagram marked in the datasheet of LED.

3

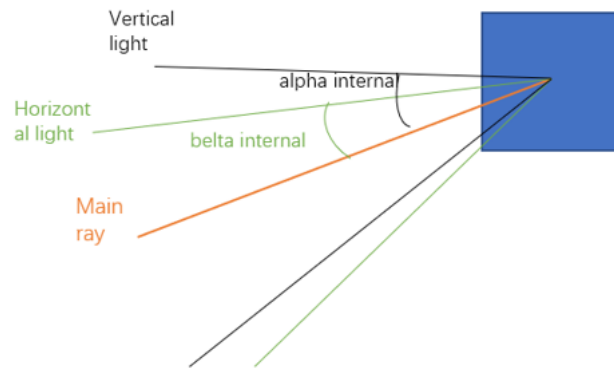


FIGURE 3: the schema of horizontal light and vertical light for each pixel

In the figure 4, we show the number of the LED array. In order to allow the light fields generated by the nine LEDs to spread on the ground, the angle between their main axes needs to be introduced. The name of the angles ($d\theta$, and $d\theta_{perp}$) are as same as the name in the matlab main program. The manufacturer can also change those angle if they want.

We have introduced four important angle $\alpha_{internal}$, $\beta_{internal}$, $d\theta$, and $d\theta_{perp}$ we can also get the value of height and angle of inclination by nucelo, which will be introduced at the following module. Now we can build a mathmatial model to study the lighting area of the target plan.

5

We build the coordinate system as shown above at figure 5, and those marked angles and lengths allow us to better understand the matlab function coordinate , this function lets us to get the coordinates of four key points. Then we use the matlab function drawarea, which allow us to draw the illuminated area by using the coordinate of four key points.

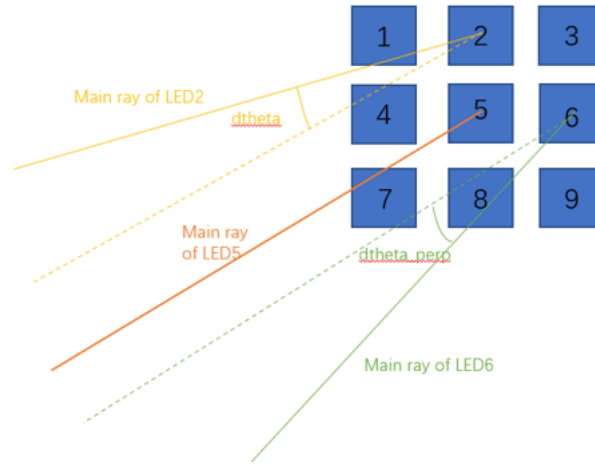


FIGURE 4: the angle introduced by orientation of different LEDs

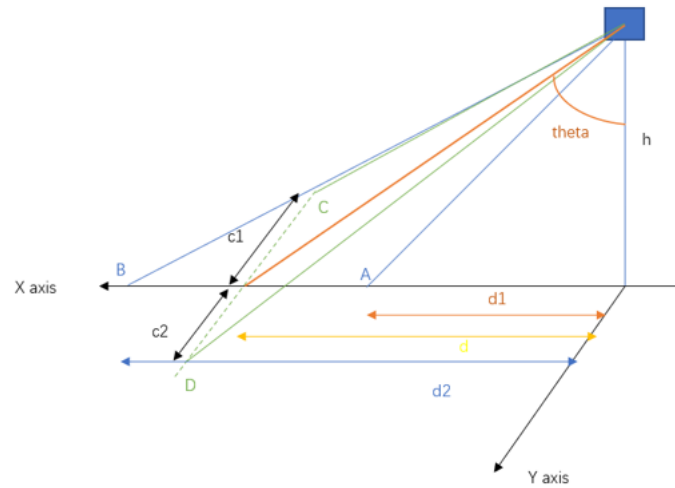


FIGURE 5: the angle introduced by orientation of different LEDs

It should be noted that in order to more conveniently represent this area, we made a reasonable approximation, basing on these four points, we draw the illuminated ellipse area on the ground (the ellipse does not necessarily pass through two points C, D).

So far, we have successfully shown the area illuminated by the center LED5 on the ground. Next, I will briefly introduce how we find the illuminated area generated by all nine LEDs on the ground in the Matlab program. For the LED2 and LED7, we can use the same way as LED5, but only change the angle when we call the function coordinate. Then we calculate the coordinate of LED4 and LED6, they share the same X coordinate with LED5, so we only need to add a modification of Y for them, this modification is introduced by the $d\theta_{perp}$. The same solution for the coordinates of other LEDs.

In the simulation process, the first thing we need to do is deciding the value of $\alpha_{internal}$, $\beta_{internal}$, $d\theta$, and $d\theta_{perp}$. Here we choose $\alpha_{internal} = 7^\circ$, $\beta_{internal} = 7^\circ$, $d\theta = 10^\circ$ and $d\theta_{perp} = 10^\circ$. we change the h and angle to observe whether the area illuminated by the LED light on the ground.

The results of simulation are shown at figure 6 and figure 7, we can see that the gap between the nine ellipses only exists when the height and angle are both small, and its size is negligible compared to the illuminated area, so we can think that the smart head-mounted led light we

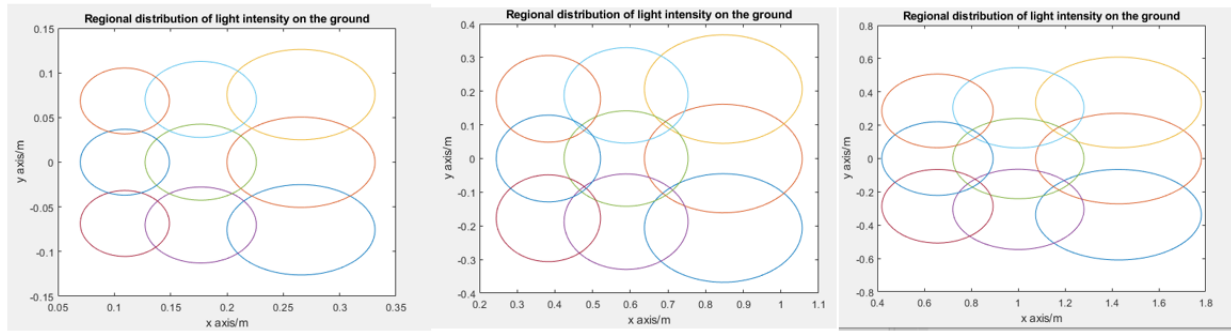


FIGURE 6: the regional distribution of light intensity on the ground when $\theta = 30^\circ$ and $h=0.3\text{m}, 1\text{m}$ and 1.7m (from left to right)

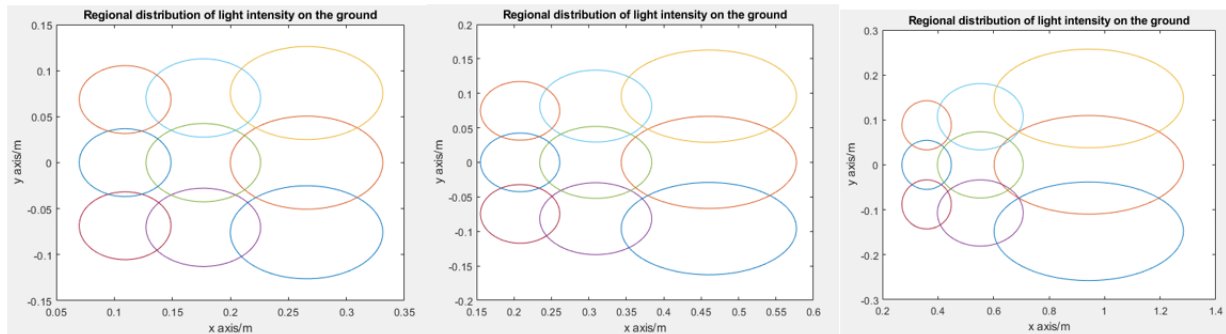


FIGURE 7: the regional distribution of light intensity on the ground when $h=0.3\text{m}$ and $\theta = 30^\circ, 45^\circ, 60^\circ$ (from left to right)

designed can produce a lighting area on the ground without gaps. It is worth noting that in the following calculations, we have ignored the overlap between different ellipses, otherwise the calculation will be too complicated.

3.2 Control the intensity of LED lights

The way we choose to control the luminous intensity of the LED lamp is to change the duty cycle of the PWM signal. According to the course, we have known that the duty cycle is proportional to the intensity of the light emitted by LED, which can be explained by the figure 8.

The mean value of signal $s(t)$ is :

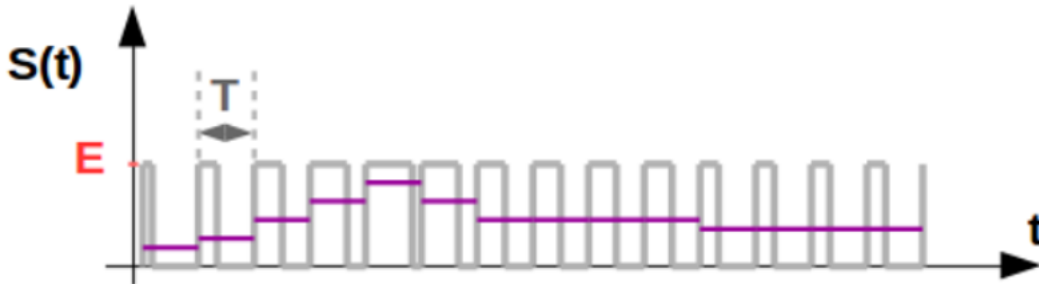


FIGURE 8: the relationship between duty circle and mean value of signal

$$s(t) = \int_0^T s(t)dt = \tau * E \quad (1)$$

It's obvious the greater the duty cycle, the greater the average signal. In our program, we use the sign I_{max} and I_{min} to represent the maximum luminous intensity and minimum luminous intensity, they are corresponding to the duty circle $\tau_{max} = 1$ and $\tau_{min} = 0.01$. The I_{max} and I_{min} can be get at the datasheet of LED, in our simulation, we will use $I_{max} = 28500mcd$ and $I_{min} = 14250mcd$. If the manufacturer changes the type of LED, they need to change those two values at the matlab function.

4 Photodiode detection module

4.1 The physical composition of the detection system

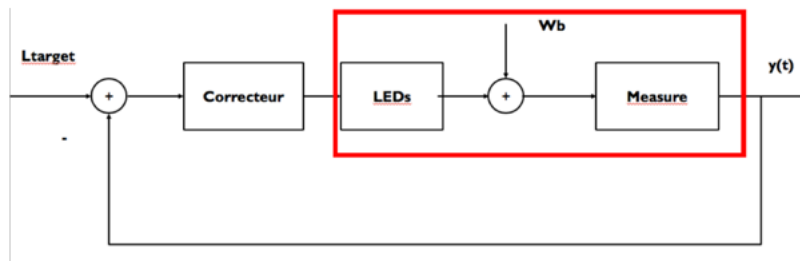


FIGURE 9: Detection system

For the detection system, for the convenience of calculation, we also use 9 photodiodes for measurement. The arrangement of these nine photodiodes is consistent with the LED, which forms a $3 * 3$ matrix.

We guarantee that the detection area of each photodiode is exactly the same as the illumina-

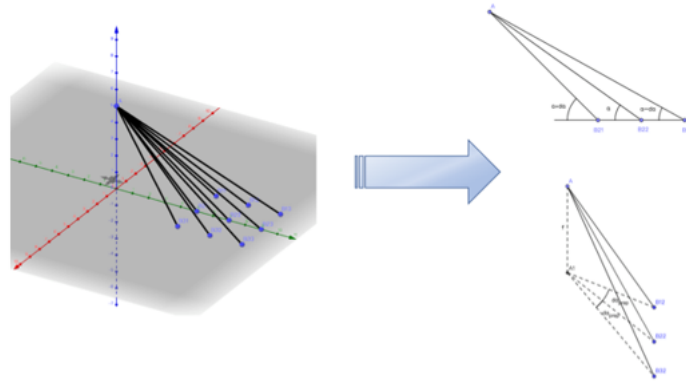


FIGURE 10: arrange of Photodiodes

tion area of the LED. This will greatly simplify our calculations in the subsequent judgment and regulation process.

For photodiode, we choose SFH203 of OSRAM as our experimental equipment. The spectral sensitivity range is 400nm-1100nm, the photodiode's photosensitive area is 1 square millimeter, and the quantum efficiency is 0.90. Through the detected photocurrent, we can calculate the number of electrons, and with the help of quantum efficiency we can calculate the number of photons, that is, we can get the received light flux. In fact, we can also use the sensitivity parameters provided in the SFH230 technical manual to directly calculate the optical power from the photocurrent. I will show some of the technical manuals of SFH230 as follows 11

In addition, our detection system consists of photodiode and lens group, which is Flux collec-

Spectral range of sensitivity	(typ)	$\lambda_{10\%}$	(typ) 400 ... 1100	nm
Radiant sensitive area	(typ)	A	1.00	mm ²
Dimensions of radiant sensitive area	(typ)	L x W	1 x 1	mm x mm
Half angle	(typ)	φ	± 20	°
Dark current ($V_R = 20$ V)	(typ (max))	I_R	1 (≤ 5)	nA
Spectral sensitivity of the chip	(typ)	$S_{\lambda \text{ typ}}$	0.62	A / W
Quantum yield of the chip ($\lambda = 850$ nm)	(typ)	η	0.90	Electro ns /Photon

FIGURE 11: the data sheet of the Photodiode SFH230

tion system.

Through this system we can calculate the L of the target area from the received luminous flux. We believe that the surface area of the lens is equivalent to the photosensitive area of the photodiode, which is 1 square millimeter. We placed the lens group close to the photodiode, so the distance from the target area to the lens is approximately equal to the distance from the target area to the photodiode. We assume that the transmission of the lens group is 1, so we can calculate the L of the target area by the following formula :

$$F_{det} = T_{opt} L_{obj} \frac{S_{obj} S_{opt}}{d_{obj-opt}^2} \Rightarrow L_{obj} = F_{det} \frac{d_{obj-opt}^2}{T_{opt} S_{obj} S_{opt}} \quad (2)$$

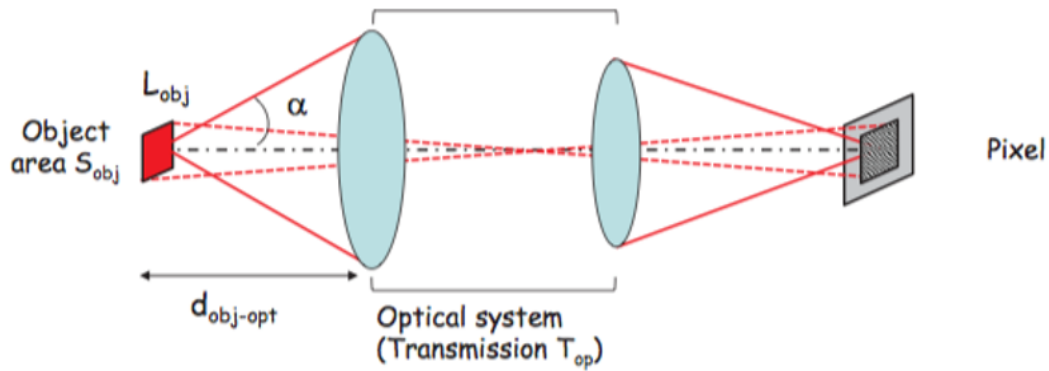


FIGURE 12: Flux collect system

Because of the lack of a hardware system, we use matlab to randomly generate photocurrents, and then calculate the final L of the target area.

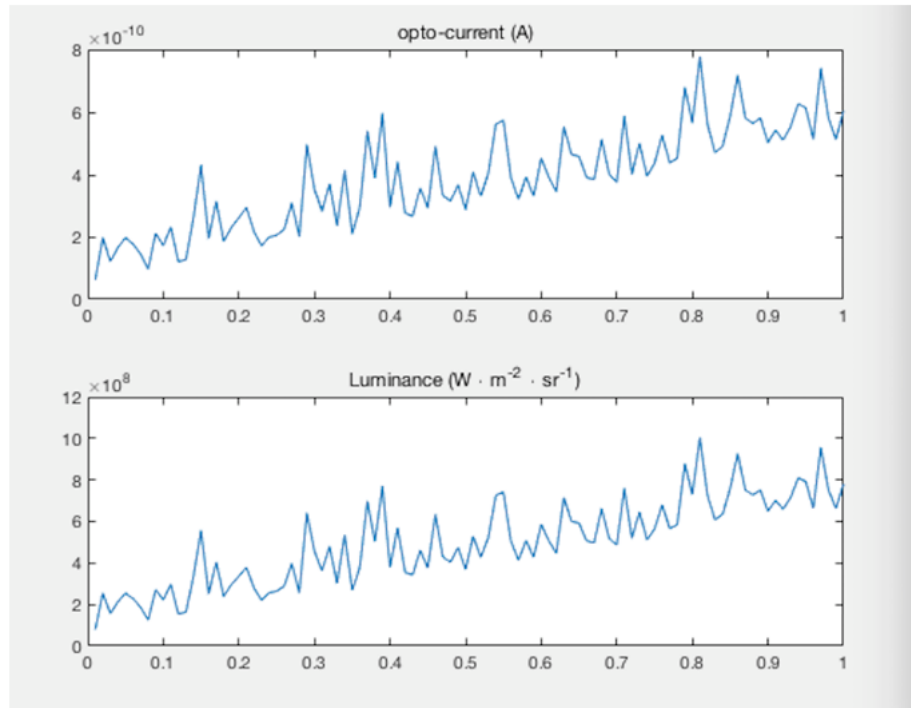


FIGURE 13: The Luminance calculated from the photo-current

4.2 Low-pass filter system

In fact, we need to adjust the Luminance of the target area by led, so that the Luminance of the target area maintains a constant illumination. But it is worth noting that this adjustment does not require a very high response rate, we only need to keep the adjustment time on the order of seconds. Because it is not only very difficult to adjust the LED more sensitively, it also has a bad effect on the human eye. Therefore, our idea is to repeatedly measure the Luminance of the target area within 1 second. Then, through a low-pass filter process, the processed values are then averaged, and the final result is fed back to the feedback system to control the led. The

reason for using low-pass filtering is that there will be some sudden changes in practice, and these changes are extremely rapid, so in our adjustment process, this change is what we want to ignore. For example, there is a sudden lightning outside the window, or a car with lighting passes by quickly. We do not want the LED to be readjusted because of such rapid changes. Therefore, we use a low-pass filter to process the signal. The following uses a photodiode as an example to show the results :

In the end, we took the measured values of the 9 areas through low-pass filtering and took

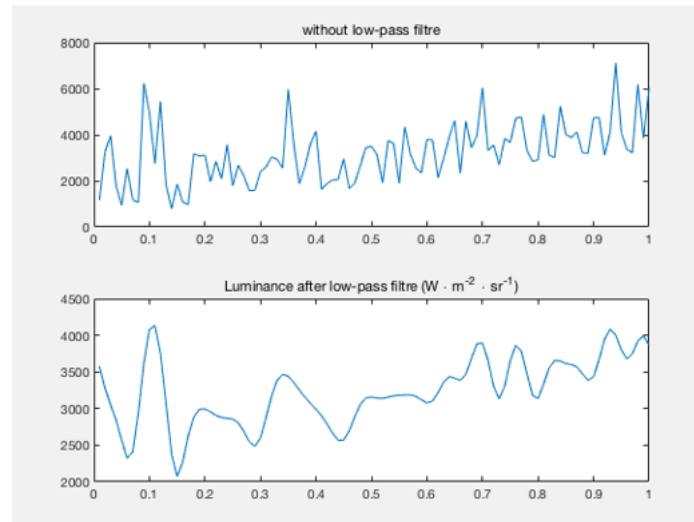


FIGURE 14: The function of the low-pass filter

their average. we got :

```

L_m =
1.0e+03 *
3.2293  2.6581  2.6283  2.8850  2.4572  2.5054  2.9971  2.7470  2.6590

```

FIGURE 15: Flux collect system

5 Feedback correction module

Our ultimate goal is to keep the Luminance of the target area around a constant target value by adjusting the light intensity of the LED. For this we built the system as shown below

Whenever the measurement system completes a measurement, the data is fed back to the correction system, and the correction system adjusts the intensity of the LED by calculation. Then the measurement system performs measurement again and gives data feedback. Until the deviation between the final measured value and the target value is within the tolerance range we set, the entire system tends to be stable and no longer changes. It is worth noting that we change the intensity of the LED by changing the duty cycle of the LED power supply. We believe that the feedback system should be implemented through timed interrupts, which will be repeated every certain time. At the same time, the detection and measurement system runs repeatedly in the main program, continuously providing feedback data. However, due to the

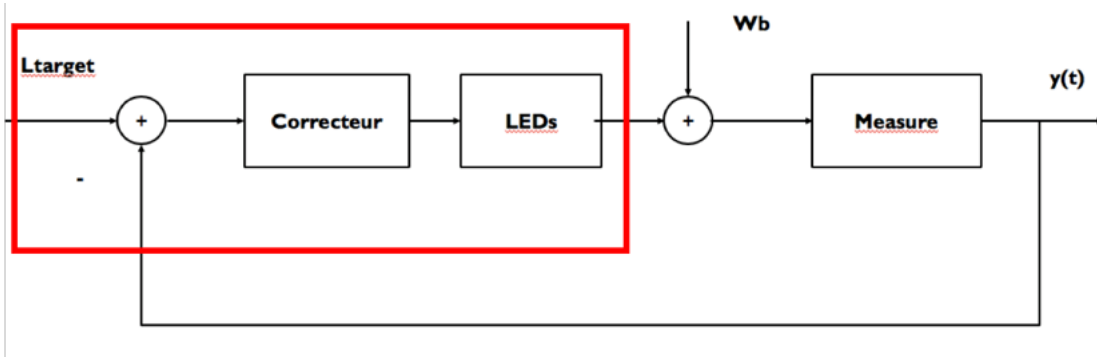


FIGURE 16: Feedback correction module

lack of hardware, it is difficult for us to complete repeated measurements, so we can directly calculate the Luminance value of the target area after changing the intensity of the LED each time through calculation. We believe that LED is a Lambertian light source. The lighting model of the ground receiving LED is Lambert diffuse reflection module :

$$I_{diff} = k_d I_{led} \cos(\theta) \tag{3}$$

where we assume $k_d = 1$.

From this we can perform simulation calculations. We can calculate the noise Luminance according to the current light intensity value of the LED and the current measured value of the target area. Then we change the light intensity of the LED, and then add the Luminance of the noise to calculate the Luminance of the target area (as we mentioned, in reality this step should be replaced with a measurement). Below we only show the correction result of a target area

We set the target Luminance value to 2000. You can see that the final Luminance of the L

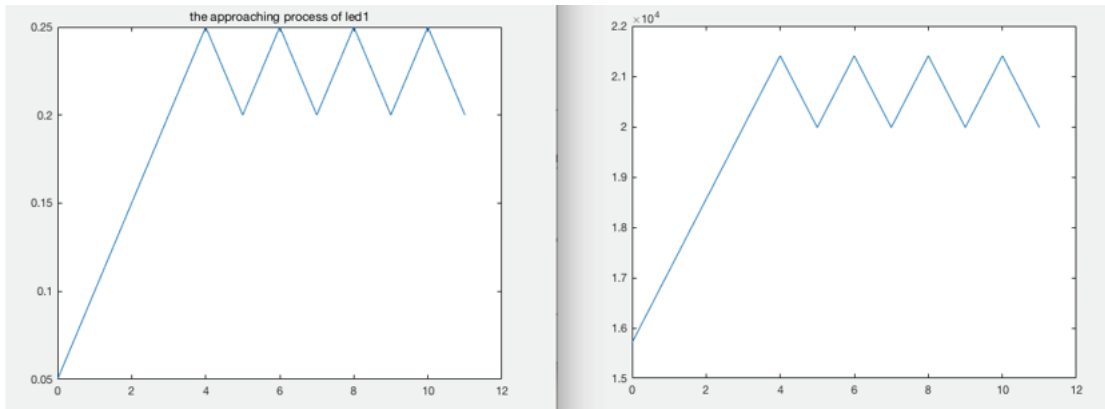


FIGURE 17: The correction process

target area bounces around 2000 (the reason for the bounce is because other target areas are still being adjusted). In the end we give the Luminance of 9 regions

You can see that it is very close to our preset value.

```

L_mm =
  1.0e+04 *
  1.8667  1.9983  2.1197  1.9081  1.9510  1.8618  1.9578  2.0330  2.1097

```

FIGURE 18: The correction process

6 Automatic distance and angle measurement module

For this module, I have done several experiments last year, so I have some real data and I think they are helpful to this project.

6.1 Automatic distance measurement

The Taking into account the existing equipment in the laboratory, I chose the Sharp GP2Y0A41SK0F infrared laser distance meter, combined with the NUCLEO L476RG micro-controller development board produced by STMicroelectronics for experimental verification. For our project, we need to choose a laser distance meter with better performance, because it can only measure the maximum height about 30cm. For the new distance meter, the manufacturer only needs to experiment according to the same method, and then replace the data in the function calibration_laser.

We used the mbed online compiler to control the infrared laser distance meter to make the measurement. At the same time, we used the serial monitored software on the computer to receive the position information at this time. Then we used Matlab to draw the distance-digital information relationship curve.

Because the laser distance meter applies the principle of triangulation, its measured value is not linear with the distance within the effective measurement range. By looking at the data sheet, the further the distance, the smaller the resulting digital value. And when the measured distance is less than the minimum measured distance, the measured value drops sharply, so the output data may be confused. When applying the infrared laser range finder, be sure to pay attention to the measurement range allowed.

In order to study the relationship between the obtained digital value and the distance, we carried out experimental verification to measure the corresponding data values at 4, 6, 8, 10, 15, 20, 25, 30 cm, and each group was measured ten times.

We can see from the above measurement that after repeated measurements, the standard deviation is very small, so the experimental results are highly reliable. We imported the data measured by the above experiment into matlab, used the polynomial fitting function to draw the detected value and distance curve which is represented by the solid blue line in the following figure. The red hollow point indicates the average value obtained by the actual measurement in the experiment.

By analyzing the curve, we can see that as the distance increases, the obtained value gradually decreases, and the two are nonlinear. After the distance is greater than 30 cm, the value becomes very slow with the distance, so it is not practical. In the experiment, when the distance is within 4cm, the distance increases, the value increases. And the maximum value exists at 3cm, so the data in this segment is not available, otherwise each measured value corresponds to two distances, and the experimental results will be confused.

time\distance (cm)	4	6	8	10	15	20	25	30
1	0.8608	0.6022	0.4535	0.3565	0.2281	0.1719	0.1211	0.1114
2	0.841	0.5844	0.4725	0.3587	0.2278	0.1751	0.127	0.1084
3	0.8364	0.5846	0.4505	0.356	0.2278	0.1641	0.1272	0.1082
4	0.8537	0.5834	0.4532	0.3575	0.2283	0.1643	0.1211	0.1111
5	0.8364	0.5832	0.4535	0.3739	0.2288	0.1751	0.1294	0.1038
6	0.8374	0.5844	0.4497	0.3565	0.2274	0.1692	0.127	0.1079
7	0.8361	0.5854	0.4518	0.3573	0.2278	0.1685	0.125	0.1092
8	0.8371	0.5851	0.453	0.3556	0.2288	0.1748	0.1223	0.1052
9	0.8364	0.5832	0.4535	0.358	0.2283	0.17	0.1221	0.1065
10	0.8836	0.5844	0.4508	0.3582	0.2278	0.1678	0.1263	0.1114
ave/cm	0.84589	0.58603	0.4542	0.35882	0.22809	0.17008	0.12485	0.10831
Standard deviation	0.01579	0.00573	0.00658	0.00539	0.00046	0.00413	0.00298	0.00261

FIGURE 19: The relationship between distance and the numerical value

6.2 Automatic angle measurement

Taking into account the existing equipment in the laboratory, we chose the MPU9250 nine-axis attitude sensor chip which produced by InvenSense, use the NUCLEO F401RE microcontroller development board for experimental verification.

Similar to the structure of the laser ranging system, we used the mbed online compiler to control the microcontroller to realize the calibration initialization and to realize the data reading settings of the MPU9250. The computer reads the data into the matlab software through the serial port, converts the data type into a floating point number type, and then the data can be applied to get the angle.

The VDD pin on the MPU9250 chip is connected to 3.3V, SDA and SCL are connected to an 8.2 k pull-up resistor, which are pulled up to 3.3V, and then connected to the SDA and SCL interfaces of the MCU. On the NUCLEO F401RE MCU, the corresponding pins are D14 and D15. The AD0 and GND of the MPU9250 are connected to the GND of the microcontroller to achieve grounding. The AD0 is grounded here. It should be noted that when using a single-chip socket, each corner solder must be fixed. Otherwise, the chip sensor I2C address will float and the attitude angle information cannot be obtained normally. For the mbed compiler code of the angle auto-acquisition module, we made some modifications based on the library files contributed by Mr. Kris Winer.

The MPU9250.h file contains the address assignments, initialization settings, and various functions used for calibration, data processing, and data fusion. Sub-functions include calibration, data reception, data fusion, and quaternion calculations. The correction of accelerometers, magnetometers and gyroscopes, and the fusion of their data are difficult. Why do we need calibration and data fusion? Because acceleration and magnetometers have high frequency noise,

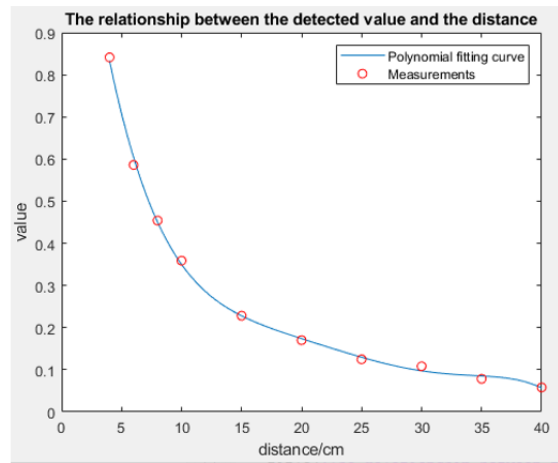


FIGURE 20: the relationship between the detected value and the distance

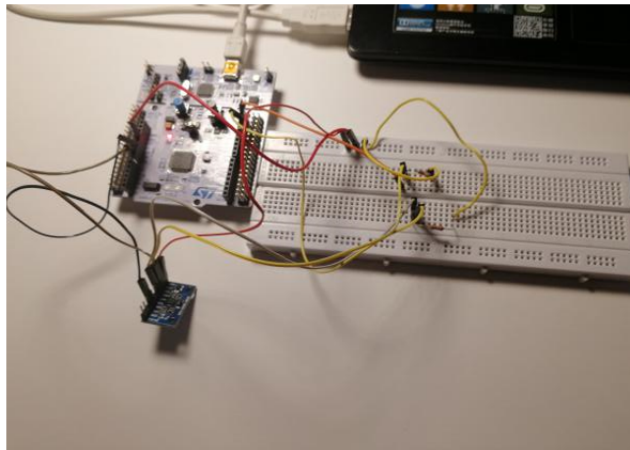


FIGURE 21: The setup of angle measurement module

low pass filtering is required. If the acceleration and magnetometer signals are regarded as audio signals, their signals will have many interference glitches, that is, their instantaneous value accuracy is low, and the calculated attitude stability is insufficient, and it will be in an oscillating state. Although the attitude direction is correct in the long run, the instantaneous value floats too much.

The gyroscope has low-frequency noise, so high-pass filtering is required, that is, the angular velocity obtained at each moment is relatively accurate, and the rotation angle (attitude) can be obtained by integrating, but the integral accumulates the error, so when the integration time is too long, the attitude information will be greatly distorted. Acceleration/magnetometers and gyroscopes complement each other in the frequency domain and combine the data from these three sensors to improve accuracy and system dynamics. The above is the meaning of the data fusion performed by the MahonyMadgwick filter algorithm.

In the main program, set I2C communication to fast mode, corresponding to 400KHz frequency. The current clock frequency of the computer is output on the computer screen because the clock frequency has an effect on the calibration result. Then output the address of the current MPU9250's *WHO_AM_I* register to verify whether the MPU9250 has been connected to the computer as expected. If the connection is successful, the success message is output, otherwise the output connection fails. After the connection is successful, reset the registers and perform a self-test, calibration, and prepare for subsequent loop measurements. Wait 0.5 seconds to

call `initAK8963` and `initMPU9250`, and display the current working mode information of the magnetometer on the computer.

After completing the above preparation work, we performed the real-time measurement in the while loop to obtain the attitude angle information of the current sensor chip. We can call the function to read the original data in the gyroscope, accelerometer and magnetometer and then save it. We performe the data fusion in real time through the Madgwick QuaternionUpdate function and obtained the quaternions. The final required angular output is obtained by the quaternion through mathematical calculation.

Monitoring the output through the Tera term software, we can get the information necessary.

```

CPU SystemCoreClock is 84000000 Hz
I AM 0x73
I SHOULD BE 0x73
MPU9250 WHO_AM_I is 0x73
MPU9250 is online...

ax = -318.481445 ay = 0.427246 az = 1024.841309 mg
gx = -0.565464 gy = -0.114609 gz = 0.022096 deg/s
mx = -483.763916 my = 155.549835 mz = 186.010986 mG
temperature = 29.095966 C
q0 = 0.823809
q1 = -0.083191
q2 = 0.125446
q3 = 0.546517
Yaw, Pitch, Roll: 53.321301 17.314568 0.003048
average rate = 3701.692871

ax = -222.717285 ay = 717.224121 az = 717.468262 mg
gx = 0.433987 gy = -0.076462 gz = 0.258607 deg/s
mx = -733.043579 my = 98.592606 mz = 60.722214 mG
temperature = 29.455387 C
q0 = 0.678881
q1 = 0.199799
q2 = 0.338646
q3 = 0.620097
Yaw, Pitch, Roll: 76.106300 12.240229 45.019440
average rate = 3713.993652

ax = -234.313965 ay = 981.506348 az = 152.465820 mg
gx = -0.649387 gy = 0.373673 gz = -0.946837 deg/s
mx = -786.569885 my = 118.604584 mz = -82.254410 mG
temperature = 30.300026 C
q0 = 0.460235
q1 = 0.273355
q2 = 0.585180
q3 = 0.609118
Yaw, Pitch, Roll: 102.064507 11.866376 80.252823
average rate = 3701.538330

```

FIGURE 22: Read the information measured by the chip with Tera term software

Then run the matlab receiving program, we can get the data. We can see from the experimental result graph that the receiving program runs successfully, and the position information of the MPU9250 sensor can be obtained at this time. The data we need is only from the perspective of roll, because our system is not suitable for user who tilt their head.

7 Data transmission module

In this part, we will explain how the data be transmitted in our system. We draw a schema to help the reader better understand it. Unfortunately, due to time constraints and limitations of experimental conditions, we cannot actually verify this part.

We can see how the data transmit between different modules according to figure 23, and we will explain them separately.

First, we start from the Nucleo F401re. I2C communication protocol is used between MPU9250 and F401re, SDA means Serial data line, it is used to transmit data, and SCL means Serial clock line, it is used to synchronize data transmission and reception. We can use F401re to receive the roll angle data under this communication protocol, we can also get the current of photodiode 5,6,7,8,9. Those data are all double type data, we send them to the computer, then we use matlab to receive those data. But the data received by matlab is not double type, it's ASCII, so we need to convert it, and we also need to pay attention to the end bit, we use the matlab function analog to realize this process.

We send the of each LED to F401RE, then use those value to control the duty circle of each LED lights, this data is between 0 and 1, so 8bits is enough. As for the L476RG, we use the Bluetooth to get the expected luminance of target surface which is inputted by the user at their smartphone. We get a value h between 0 and 1 which include distance information by using laser distance meter, we can also get the current of photodiode 1,2,3,4. Then we send those value to the matlab.

The last technical problem is that we use two nucleo because there are no enough analog input pins for our 9 photodiodes. The two MCU boards are connected to the two ports of the computer. In the main function of matlab, we need to open one of the ports, read in the data, and then close it. Then turning on the other, read in the data transmitted by the other MCU board, and loop in turn. This may cause the program to run slower, and needs to be improved in future projects.

After we get all that information from the outside, now we are able to do the calculation which are mentioned in the previous modules.

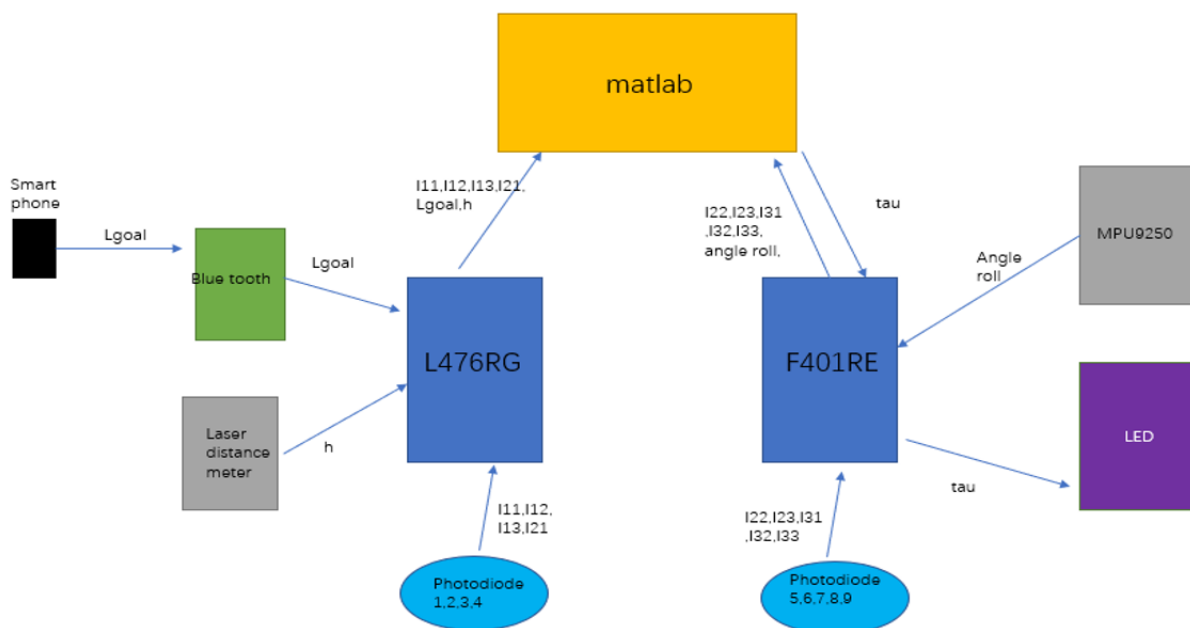


FIGURE 23: The schema of information transmission path

8 Interface graphique utilisateur

L'interface graphique à été réalisé avec le système développé par le MIT : APP Inventor disponible sur le site : <https://appinventor.mit.edu/>.

Le système se présente par une partie bloc de code écrit de manière simplifié et une partie design ou on créer l'architecture de notre application. Nous allons détailler les différents menues ainsi que le choix d'une application par téléphone.

8.1 Choix de l'application

Nous devons réaliser une interface graphique accessible à tous. Notre projet à une finalité pour être utilisé partout, dans un bureau, comme dans la nature avec son programme de course. Nous devons donc avoir une interface qui soit facilement accessible pour adapter l'intensité voulu par l'utilisateur. De plus si l'utilisateur veux facilement changer la luminosité ambiante, il est préférable qu'il puisse faire le changement par un moyen facile, rapide.

Pour pouvoir réaliser toutes ces conditions, une application mobile nous paraissait satisfaisante. De plus l'appareil nous permettant d'effectuer la connexion est relativement facilement programmable. Nous avons donc opté pour une connexion de type bluetooth.

8.2 Recherche du périphérique bluetooth

Une des premières étapes importantes lors du démarrage de l'application est de vérifié la connections bluetooth. Si ce dernier est inactive, une notification apparaît alors pour nous demander de l'activer.

La deuxième étapes consiste à se connecté au bluetooth du système de LED. Pour cela nous choisissons un module HC-05 qui fonctionne en 3.3V et qui permet de recevoir et d'émettre des informations par bluetooth. Sur l'écran de l'application, en cliquant sur l'image principale, nous avons accès à la liste de tous les bluetooths environnant disponible. Ainsi on peut sélectionner celui associé à notre système.

Dansle code, la première boucle when concerne la notivication de présence ou d'absence

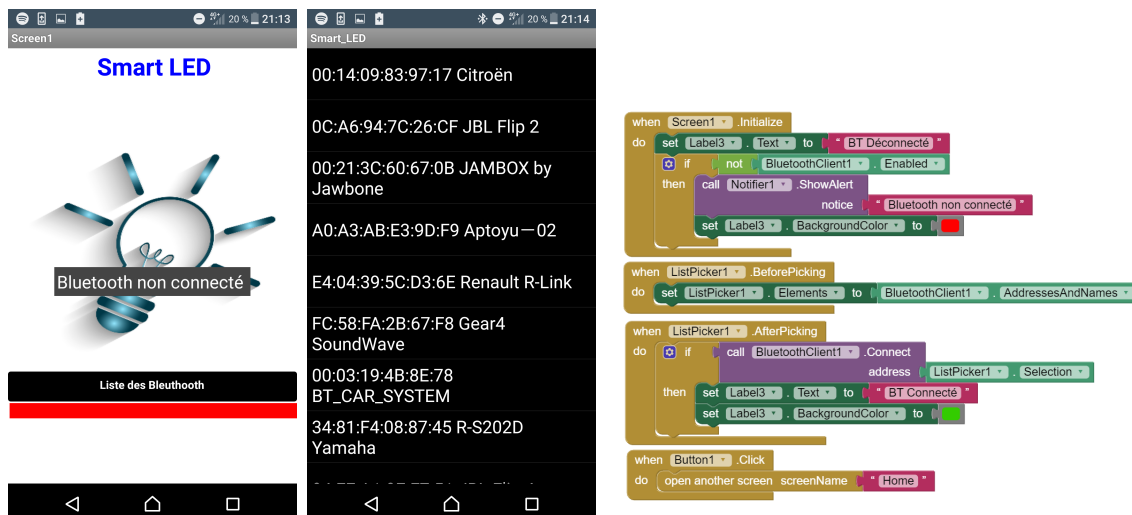


FIGURE 24: De gauche à droite : page principale avec notification bleutooth, liste des périphériques environnants, code block du programme associé à cette page

de bluetooth. Si ce dernier est éteint alors : la notification apparaît et sous le bouton liste bluetooth on voit un message qui nous avertit de l'état de bluetooth.

Dans la deuxième boucle on affiche la liste des bluetooth. Dans la troisième, lorsqu'on clique sur "liste des bluetooth" et qu'on sélectionne celui correspondant, l'état de la notification change. La dernière boucle permet simplement d'accéder à l'écran suivant.

8.3 Sélection du mode et envoie d'information

Sur la deuxième page, on peut sélectionner le mode de fonctionnement que l'on veut. L'avantage ici c'est que pour chaque mode, la valeur dernièrement enregistré est maintenue en mémoire, voilà pourquoi on sépare les différents modes : pour garder une répétabilité entre les diverses utilisation. On présente si après l'écran et le code associé. Il s'agit uniquement de renvoie vers les pages décrites dans la sous partie suivante.

De même l'écran du mode est constitué comme sur la photo 26. Un case permet de rentrer la

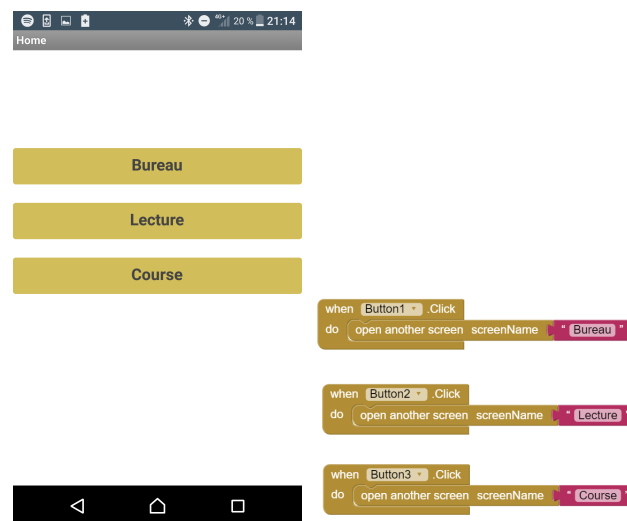


FIGURE 25: De gauche à droite : page de sélection des modes, code block du programme associé à cette page

valeur d'intensité que l'on veut en pourcentage. Le pourcentage correspond à un niveau d'intensité disponible entre 0% la valeur correspondant à des LEDs éteintes et 100% à la luminosité maximal pouvant être atteinte par les LEDs. Cela correspond donc à un niveau de luminosité que l'utilisateur sait interpréter comme cela et ne requiert aucune connaissance technique. Cette intensité est stocké dans une variable qui est ensuite envoyé par l'appelle du bluetooth

8.4 Installation du code sur smartphone

MIT APP Inventor, permet de générer automatiquement un fichier APK que l'on peut enregistrer sur son téléphone depuis l'ordinateur. Ainsi l'application est facilement transportable et s'adapte à tous les utilisateurs en très peu de temps.

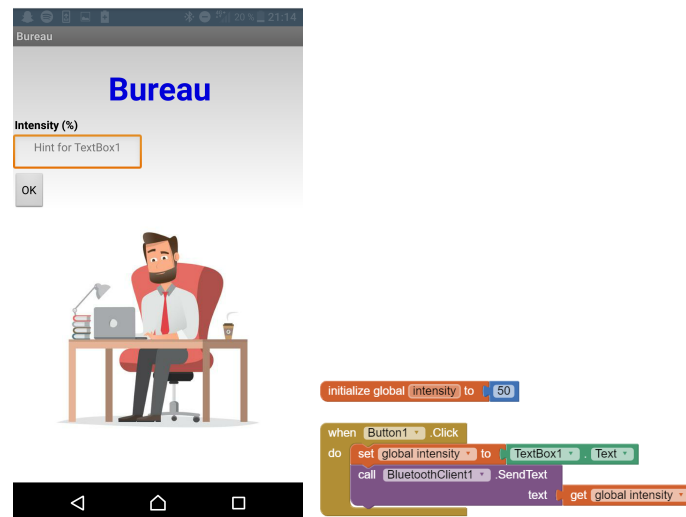


FIGURE 26: De gauche à droite : page de controle de l'intensité lumineuse, code block du programme associé à cette page

9 Design du Circuit

L'ensemble du circuit a été fait sous Fritzing qui est un logiciel de montage, schémas électriques et simulations. On peut le visualiser figure 27 On retrouve ici plusieurs des composants

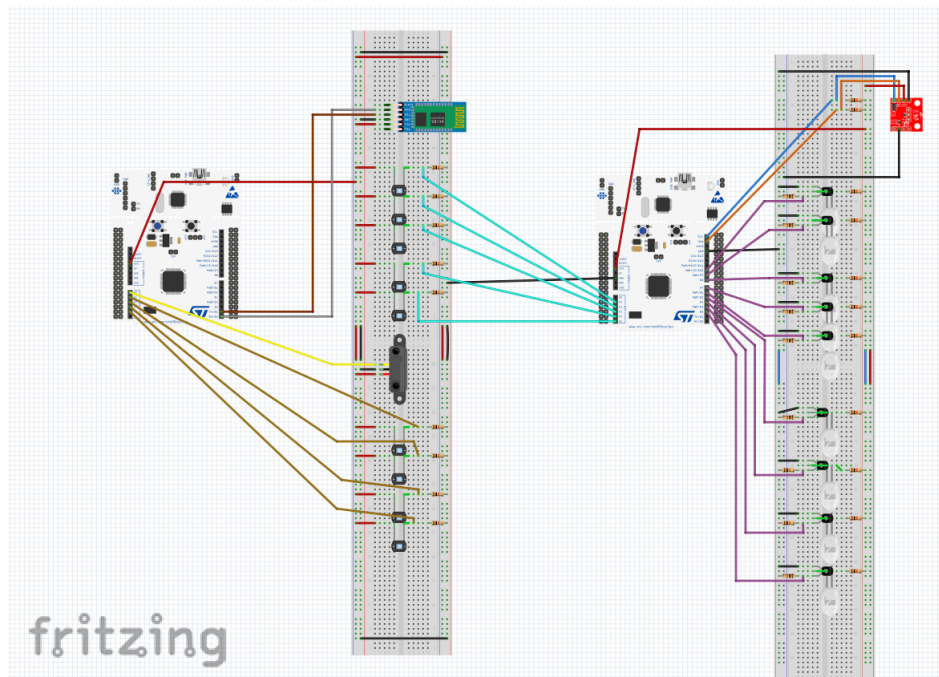


FIGURE 27: Montage électrique de notre système

utilisés dans les parties précédente cela est détaillé dans le tableau 28.

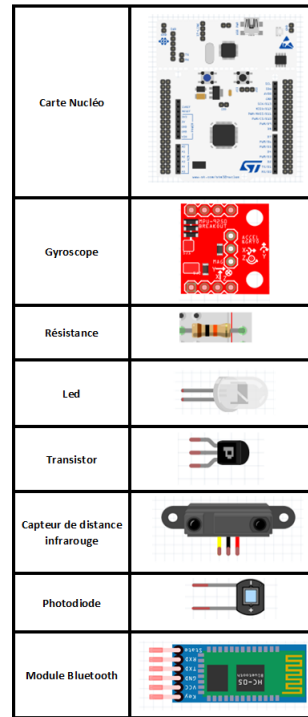


FIGURE 28: Ensemble des éléments du système

10 Conclusion and amélioration

Ce premier prototype permet donc de satisfaire un bon nombre des contraintes que l'on rappelle ici : Éclairer une scène de manière à posséder automatiquement un éclairage uniforme dans le temps et l'espace, adapté le système à plusieurs modes de fonctionnement, Analyser la luminosité d'une scène et adapté l'éclairage en fonction.

Pour cela nous avons fait plusieurs hypothèses et le choix de ne pas prendre en compte les courts événements qui n'ont pas vraiment d'importance dans l'éclairage de la scène (comme un vol d'insecte, ou une personne qui passe rapidement devant nous).

Nous avons mis en place une interface utilisateur facilement instable et ergonomique pour pouvoir contrôler notre système smart_LED.

Cependant, il est important de préciser que de nombreuses améliorations sont à apporter à notre système :

Premièrement, notre projet est assez compact, il est donc nécessaire de fusionner les deux cartes en une seule via une carte Nucléo ayant suffisamment de broches ou de passer sur un Arduino MEGA, mais les interruptions ne se déroulent pas de la même façon.

Deuxièmement, il faudrait avoir recours à une lentille à placer devant la matrice de photodiode afin d'imager sur chaque photodiode une partie bien définie de la scène que l'on analyse. Le but étant aussi de ne pas capturer la lumière d'une partie de la scène sur plusieurs photodiodes. Ce problème ne se pose pas sur l'émission de la lumière des LEDs car celle-ci est assez directive.

Troisièmement, il faudra réaliser certains tests pour améliorer notre matrice de LED et photodiode.

Enfin, il faudrait mettre tous les dispositifs dans une structure ergonomique et confortable pour l'utilisateur (que ce soit en utilisation frontale ou en utilisation fixée sur un socle).