Email Sending Project with Data Received from Light Sensor

This project basically aims to continuously monitor data obtained from a light sensor. The system will automatically react when the data received from the light sensor rises above a certain preset value. This response would be to send a notification email to a predetermined email address.

The second aspect of the project involves the continuous recording of the resulting data. Data will be received from the light sensor every 3 seconds and this data will be regularly saved in an Excel file. Along with each data point, the relevant date and time will also be automatically added to the Excel file, allowing you to easily track how the data changes over time.

Ultimately, this project will provide an automatic mechanism to monitor light sensor data and send notifications when certain conditions are met. At the same time, recording data at regular intervals will create a valuable data source that can be used for analysis or monitoring purposes in the future.

To realize the project, I decided to use the Arduino board. This choice appeared to be a convenient way to obtain sensor data and then process it in Python. It was enough to get an Arduino set with the basic components I needed to manage this process. I will use tools like the Arduino IDE and Pycharm on my computer to complete the remaining steps. In this way, I planned to establish an infrastructure where I could obtain the results I wanted by receiving sensor data via Arduino and then processing it with Python on the computer.

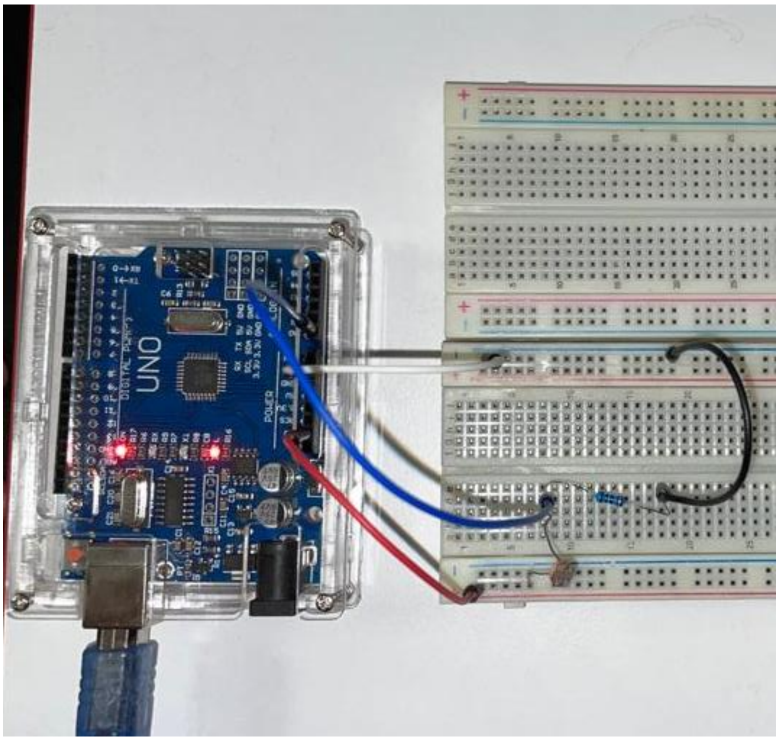
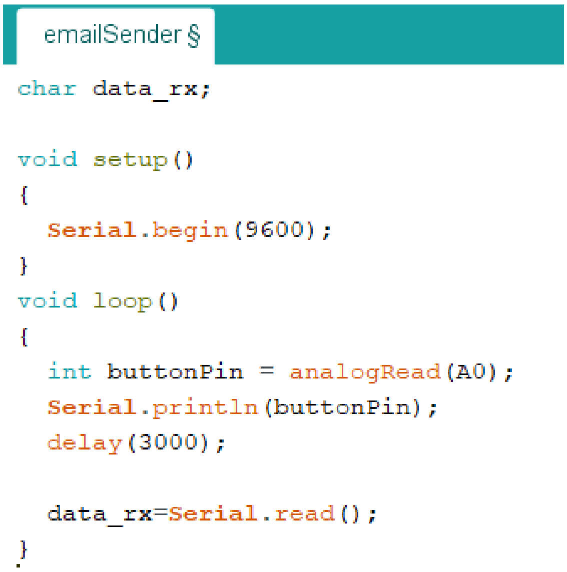


Figure 1 - The way the Arduino circuit is connected.

I connected my circuit as seen in ***Figure 1*** above. After this process was completed, the next step was to write the necessary codes to receive and process the data from the light sensor using ***the Arduino IDE***.



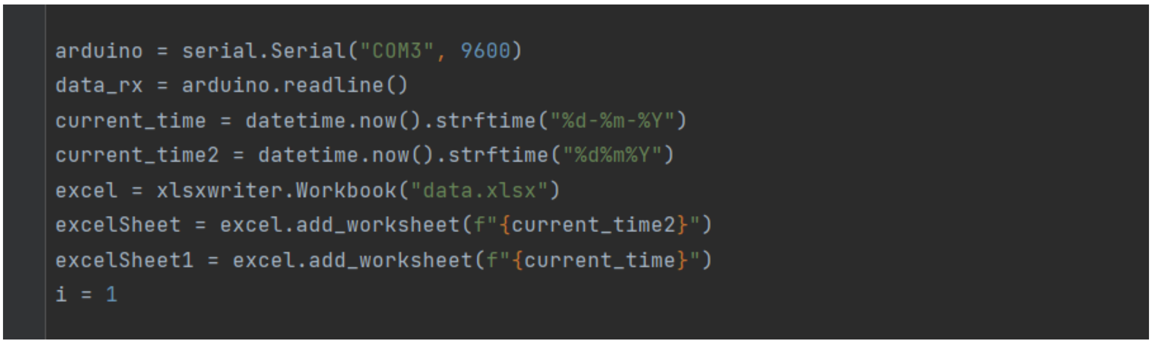
```
char data_rx;

void setup()
{
  Serial.begin(9600);
}
void loop()
{
  int buttonPin = analogRead(A0);
  Serial.println(buttonPin);
  delay(3000);

  data_rx=Serial.read();
}
```

Figure 2 - Codes that receive data from light sensor in Arduino IDE

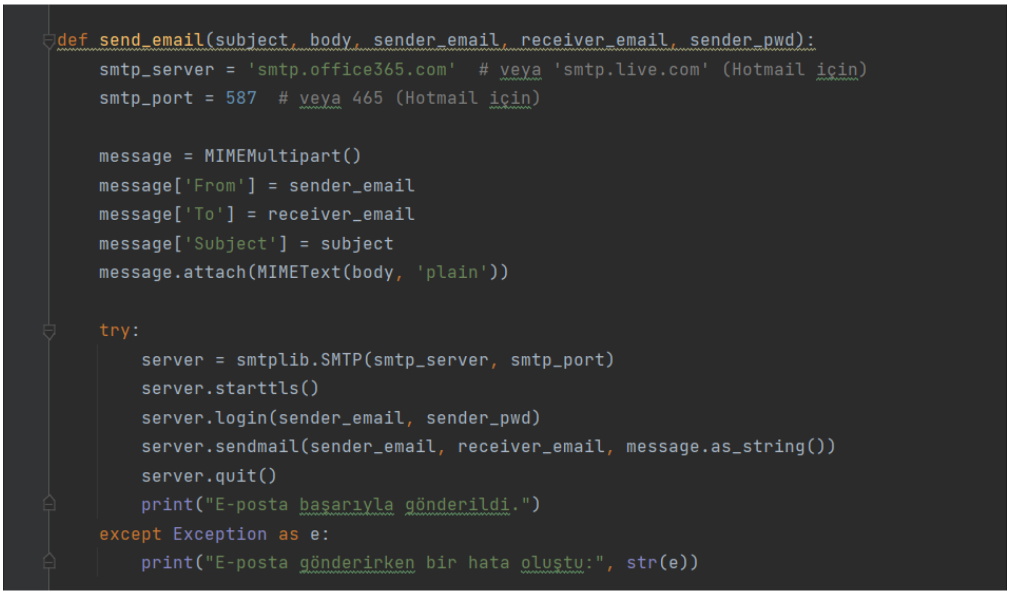
As seen in ***Figure 2*** above, I received the data from the light sensor. Then I prepared this data to transfer it to ***Pycharm***. After these operations, the operations on ***the Arduino IDE*** are completed. Now I switched to ***Pycharm*** and started writing my Python codes.



```
arduino = serial.Serial("COM3", 9600)
data_rx = arduino.readline()
current_time = datetime.now().strftime("%d-%m-%Y")
current_time2 = datetime.now().strftime("%d%m%Y")
excel = xlsxwriter.Workbook("data.xlsx")
excelSheet = excel.add_worksheet(f"{current_time2}")
excelSheet1 = excel.add_worksheet(f"{current_time}")
i = 1
```

Figure 3 - Codes for connecting Python with Arduino IDE

As seen in ***Figure 3***, I used ***the Serial command*** to connect Arduino and Python and get instant data. For this, I added the ***Serial library*** in Pycharm. In this way, I entered the port I used for Arduino and its baud value in ***the Serial command*** and connected them to each other. I uploaded the data I received to ***"arduino"*** and sent the read data to the ***"data_rx"*** integer. Apart from these, I uploaded the current computer date and time to ***"current_time"*** and ***"current_time2"***. I also opened a ***"data.xlsx"*** file into the Excel file and added the worksheets.



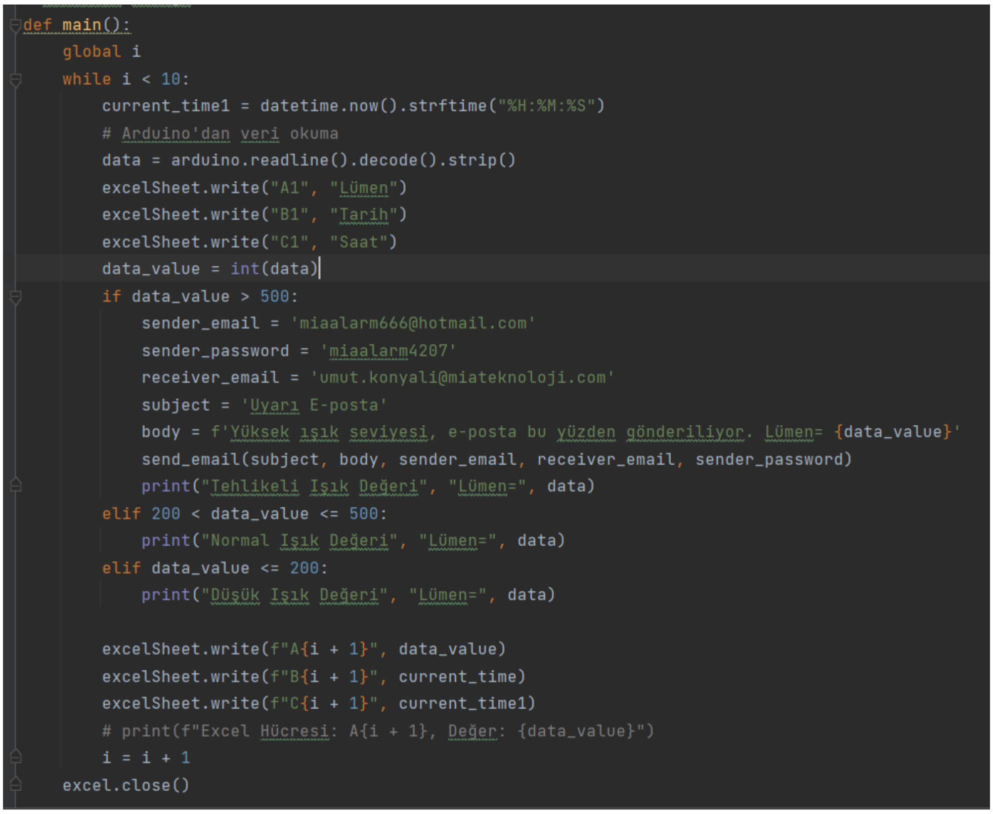
```
def send_email(subject, body, sender_email, receiver_email, sender_pwd):
    smtp_server = 'smtp.office365.com'  # veya 'smtp.live.com' (Hotmail için)
    smtp_port = 587  # veya 465 (Hotmail için)

    message = MIMEMultipart()
    message['From'] = sender_email
    message['To'] = receiver_email
    message['Subject'] = subject
    message.attach(MIMEText(body, 'plain'))

    try:
        server = smtplib.SMTP(smtp_server, smtp_port)
        server.starttls()
        server.login(sender_email, sender_pwd)
        server.sendmail(sender_email, receiver_email, message.as_string())
        server.quit()
        print("E-posta başarıyla gönderildi.")
    except Exception as e:
        print("E-posta gönderirken bir hata oluştu:", str(e))
```

Figure 4 - Mail sending codes.

As seen in ***Figure 4***, I wrote the necessary codes for sending e-mail. For these codes, I first added ***the e-mail library*** to Pycharm. Then, as seen in the figure, I entered the ***SMPT server*** and ***SMTP ports*** and completed the intermediary processes to send. I created the name ***"message"*** to write information such as sending e-mail, receiving e-mail, subject, and description, respectively. I prepared the sending process by creating a ***"Try-Exception"***.

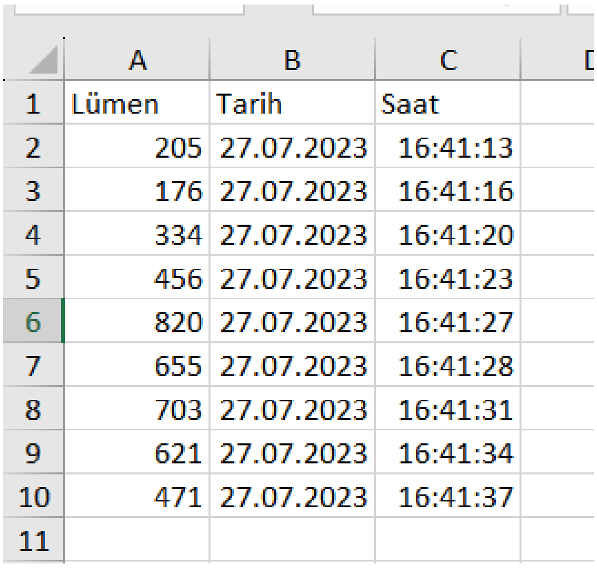


```
def main():
    global i
    while i < 10:
        current_time1 = datetime.now().strftime("%H:%M:%S")
        # Arduino'dan veri okuma
        data = arduino.readline().decode().strip()
        excelSheet.write("A1", "Lümen")
        excelSheet.write("B1", "Tarih")
        excelSheet.write("C1", "Saat")
        data_value = int(data)
        if data_value > 500:
            sender_email = 'miaalarm666@hotmail.com'
            sender_password = 'miaalarm4207'
            receiver_email = 'umut.konyali@miateknoloji.com'
            subject = 'Uyarı E-posta'
            body = f'Yüksek ışık seviyesi, e-posta bu yüzden gönderiliyor. Lümen= {data_value}'
            send_email(subject, body, sender_email, receiver_email, sender_password)
            print("Tehlikeli Işık Değeri", "Lümen=", data)
        elif 200 < data_value <= 500:
            print("Normal Işık Değeri", "Lümen=", data)
        elif data_value <= 200:
            print("Düşük Işık Değeri", "Lümen=", data)

        excelSheet.write(f"A{i + 1}", data_value)
        excelSheet.write(f"B{i + 1}", current_time)
        excelSheet.write(f"C{i + 1}", current_time1)
        # print(f"Excel Hücresi: A{i + 1}, Değer: {data_value}")
        i = i + 1
    excel.close()
```

Figure 5 - Codes for reading data, Sending mail and Printing data to Excel.

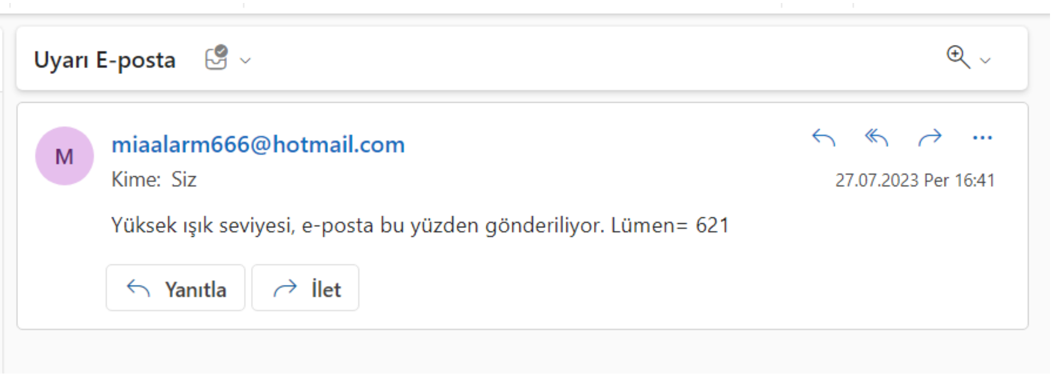
Figure 5 is the visual where all the desired things are achieved. Here, first, I put the codes for ***Excel*** into a ***"while"*** loop. Because when it entered an infinite loop, there was a problem in saving ***the Excel data*** because there was no end to the loop. By adding ***"current_time1"*** into the loop, I wanted the time to be written every time data was entered into Excel. Then, to read the data from the light sensor from Arduino, I created a ***"data"*** name and printed it. Apart from these, in the first row in Excel, I wrote the names ***"Lumen"*** in the first column, ***"Date"*** in the second column and ***"Time"*** in the third column. In this way, I was able to understand which data was in which column. Now the read data needs to be processed. For this, if the light value in ***'if'*** is less than 200 lumens, ***"low light value"*** is printed. If the light value is between 200-500 lumens, it is printed as ***"Normal light value"*** and if the lumen value is greater than 500, it is printed as ***"Dangerous light value"***. Apart from this, if the lumen value is greater than 500, to send an e-mail, the sending e-mail, the password of the sending e-mail, the recipient e-mail, the subject, and the content of the e-mail are printed and sent using ***"send_mail"***. I completed the project given to me by writing all the necessary codes.



	A	B	C
1	Lümen	Tarih	Saat
2	205	27.07.2023	16:41:13
3	176	27.07.2023	16:41:16
4	334	27.07.2023	16:41:20
5	456	27.07.2023	16:41:23
6	820	27.07.2023	16:41:27
7	655	27.07.2023	16:41:28
8	703	27.07.2023	16:41:31
9	621	27.07.2023	16:41:34
10	471	27.07.2023	16:41:37
11			

Figure 6 - Writing data in Excel file.

I printed the data to the Excel file as seen in ***Figure 6***.



Uyarı E-posta

miaalarm666@hotmail.com

Kime: Siz

27.07.2023 Per 16:41

Yüksek ışık seviyesi, e-posta bu yüzden gönderiliyor. Lümen= 621

Figure 7 - A mail sent because it exceeds 500 Lumens.

An e-mail sent when it exceeds 500 Lumens, as seen in ***Figure 7***.