# **Отчет по большой лабораторной работе на тему:**

# **Сигнальный регистратор пришедших электронных писем.**

Подготовил студент группы М24-Ш02 Китаев С.М.

Сигнальный регистратор пришедших электронных писем - это устройство интернета вещей, которое призвано включать сигнал, которым в работе выступает красный светодиод, в момент получения входящего письма на электронную почту.

Для того, чтобы выполнить данный проект необходимо решить несколько задач:

1. Создание электронной почты, с доступом для внешних приложений, в частности возможности манипуляции с сообщениями электронной почты при помощи протокола IMAP;
2. Написание кода прослойки, который способен как проводить манипуляции с электронной почтой, так и посылать сигналы на плату Arduino uno;
3. Написание рабочего кода для платы Arduino uno, который принимает сигнал с кода прослойки и выполняет пришедшую команду;
4. Сбор электросхемы.

Инструменты и оборудование, которое используется при выполнении лабораторной работы:

1. Персональный компьютер с установленными программами PyCharm 2024 и Arduino Uno и выходом в интернет;
2. Установленный Python версией выше 3.7;
3. Плата Arduino uno с проводом подключения к USB порту персонального компьютера;
4. Макетная плата;
5. 2 провода;
6. Резистор на 1 кОм;
7. Красный светодиод.

Выбор макетной платы, проводов, резистора и цвета светодиода может отличаться от примера.

## **Создание электронной почты, с доступом для внешних приложений, в частности возможности манипуляции с сообщениями электронной почты при помощи протокола IMAP.**

Для создания электронной почты было решено использовать почту mail.ru так как это отечественная почта, которая поддерживает доступ внешних приложений по протоколу IMAP. Для выполнения данной лабораторной работы был создан новый аккаунт почты с подключением внешних приложений.

Первое действие это регистрация электронной почты, что показано на рисунке 1.1

|  |
| --- |
|  |
| Рисунок 1.1 – Регистрация новой электронной почты |

Теперь электронная поста создана, стартовая страница показана на рисунке 1.2

|  |
| --- |
|  |
| Рисунок 1.2 – Стартовая страница почты |

Далее необходимо открыть доступ к почте для внешних приложений. Для этого необходимо пройти по следующий путь: в нижней левой части экрана Настройки – Все настройки – безопасность – Все настройки безопасности – Пароли для внешних приложений – Создать – Новый пароль для внешнего приложения – Настройки доступа (Полный доступ к Почте «IMAP, SMTP, POP3»). Данный процесс показан на рисунке 1.3

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | | |
| Рисунок 1.3 – Создание пароля для внешних приложений | | |

На рисунке 1.4 показано окно после создания пароля для внешних приложений. Данный пароль сохраните.

|  |
| --- |
|  |
| Рисунок 1.4 – Пароль для внешних приложений |

1. **Написание кода прослойки, который способен как проводить манипуляции с электронной почтой, так и посылать сигналы на плату Arduino uno.**

Так как код уже написан, и вам требуется только правильно только его запустить, то стоит подробнее рассказать про то, как его запускать и как он работает. Как уже было сказано ранее, необходимо чтобы у вас был установлен PyCharm 2024 (может быть установлена и другая версия, но в данном документе работы описаны для данной версии PyCharm), а также установлен python версии 3.7 или выше (работу данной программы на других версиях python не гарантируется).

Для начала необходимо создать новый проект в PyCharm, окно с созданием нового проекта показано на рисунке 2.1.

|  |
| --- |
|  |
| Рисунок 2.1 – Окно создания нового проекта |

Далее необходимо создать новый файл .py, на котором и будет написан код-прослойка. Данный код должен взаимодействовать как с платой Arduino, так и с электронной почтой. Взаимодействие кода и платы будет организовано при помощи библиотек serial и pyserial, а взаимодействие с электронной почтой будет организовано при помощи библиотеки imaplib. Imaplib устанавливается в PyCharm автоматически, а serial и pyserial необходимо дополнительно установить. Для того, чтобы это сделать необходимо перейти в правый верхний угол и зайти в settings – project –   
Python Interpreter – нажать на знак «+» как показано на рисунке 2.2 стрелкой, затем в меню поиска найти необходимые библиотеки, которые у вас не установлены и установить их.

Далее будет представлен код прослойки python.

|  |
| --- |
|  |
| Рисунок 2.2 – Окно загрузки библиотек |

import imaplib

import time

import re

import serial

# Инициализация параметров

mail\_pass = "(пароль для приложений)"

username = "(ваша почта)"

port = 'COM(номер порта)'

def checker():

imap\_server = "imap.mail.ru"

try:

imap = imaplib.IMAP4\_SSL(imap\_server)

imap.login(username, mail\_pass)

ser = serial.Serial(port, 9600)

none = "none"

# Инициализация счётчика

match = re.search(r'\d+', str(imap.select("INBOX")[1]))

changer = int(match.group())

# Работа считывателя пришедшего письма

start = time.time()

while time.time() - start < 30:

time.sleep(1) # Увеличен интервал проверки

try:

match = re.search(r'\d+', str(imap.select("INBOX")[1]))

x = int(match.group())

if x > changer:

print('Пришло новое письмо!')

time.sleep(1)

ser.write("sign".encode())

time.sleep(1)

changer = x

else:

time.sleep(1)

ser.write(none.encode())

time.sleep(1)

except imaplib.IMAP4.abort:

print("Соединение с сервером потеряно, повторное подключение...")

break

except Exception as e:

print(f"Ошибка: {e}")

finally:

try:

imap.logout()

ser.close()

except:

pass

while True:

checker()

Данный код работает следующим образом – он каждые 30 секунд подключается к почте инициализирует количество сообщений, и в течении 30 секунд проверять изменение в количестве сообщений. Если сообщений стало больше, что идентично пришедшему сообщению, то посылается сигнал на порт Arduino с обозначением «signal», если же изменений нет, то посылается сигнал «none», то есть, что изменений нет. В процессе работы кода намеренно были сделаны 2 задержки в 1 секунду, так как для обработки сигнала плате Arduino требуется время.

Постоянное переподключение к почте необходимо из-за того, что IMAP позволяет подключиться к почте на непродолжительное время 1 – 2 минуты, после чего сервер с почтой разорвет соединение, чтобы не нагружать систему, поэтому постоянное переподключение это единственный выход.

Также в код была добавлена обработка ошибок и функционал попыток переподключения в случае возникновения ошибок.

1. **Написание рабочего кода для платы Arduino uno, который принимает сигнал с кода прослойки и выполняет пришедшую команду.**

Далее необходимо написать код на Arduino и загрузить его на плату. Код должен выполнять 2 функции:

1. Считывать пришедшие с python кода сигналы;
2. В зависимости от пришедшего сигнала зажигать или не зажигать лампочку

Код для Arduino представлен ниже

int led\_pin = 3;

void setup() { Serial.begin(9600);

pinMode(led\_pin, OUTPUT);}

void loop() {

String text = Serial.readString(); if (text == String("sign")){

digitalWrite(led\_pin, HIGH); delay(2000);

} else if (text == String("none")){

digitalWrite(led\_pin, LOW); delay(2000);

}

}

Задача, которая выполняется на плате Arduino простая, поэтому код небольшой. Дополнительно замечу, что пином вывода является 3 пин, вы можете изменить пин по своему усмотрению.

1. **Сбор электросхемы.**

Ввиду простоты кода на Arduino, простой будет являться и схема. Изображение

схемы представлено на рисунке 4.1.

|  |
| --- |
|  |
| Рисунок 4.1 – Схема для платы Arduino |

**Заключение**

Если вы сделали все как сказано в данной работе, то у вас должно было все получиться. Дополнительно замечу, что отклик у светодиода не моментальный и составляет 4–5 секунд. Это из-за того, что 2 секунды задержки были выставлены намеренно, для корректной передачи сигнала на плату Arduino, секунда задержки появляется ввиду задержки между отправкой сообщения с одной почты и получением сообщения другой почтой и еще 1–2 секунды набегают из-за дополнительных задержек, связанных со скоростью интернет подключения, скоростью обработки сигнала платой, и «дальностью» порта от компьютера (при выполнении данной работы я использовал 2 переходника для подключения платы).