

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРИВОРІЗЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ
Факультет дошкільної і технологічної освіти
Кафедра загальнотехнічних дисциплін та професійного навчання

«Допущено до захисту»

Завідувач кафедри ЗТДПН

_____ Олег Цись
« ___ » _____ 2023 р.

Реєстраційний № _____
« ___ » _____ 2023 р.

**РОЗРОБКА Й ВИГОТОВЛЕННЯ МОДЕЛІ SMART HOME НА БАЗІ
ARDUINO ТА МЕТОДИКА ВИКОРИСТАННЯ ПРИ ВИВЧЕННІ
ТЕХНОЛОГІЇ У ПРОФІЛЬНІЙ ШКОЛІ**

Кваліфікаційна робота студента
групи ТОАм-22
ступінь вищої освіти магістр
спеціальності
014.10 Середня освіта (Трудове навчання
і технології)
Жадька Антона Олександровича
Керівник: к.тех.н., доц.
Кучма Олександр Іванович

Оцінка:

Національна шкала _____

Шкала ECTS _____ кількість балів _____

Голова ЕК _____

Члени ЕК _____

ЗМІСТ

ВСТУП	3
1. ЗАСТОСУВАННЯ ПРОГРАМНО-АПАРATНОЇ ПЛАТФОРМИ ARDUINO В ОСВІТНЬОМУ ПРОЦЕСІ	5
1.1 Застосування програмно-апаратної платформи Arduino в освітньому процесі.....	5
1.2 Можливості програмно-апаратної платформи Arduino в освітньому процесі	7
2. РОЗРОБКА Й ВИГОТОВЛЕННЯ МОДЕЛІ SMART HOME НА БАЗІ ARDUINO.....	10
2.1 Особливості програмно-апаратної платформи Arduino.....	10
2.2 Етапи розробки ескізного проєкту Smart Home на базі Arduino	12
2.3 Розробка й виготовлення моделі Smart Home на базі Arduino.....	24
3. МЕТОДИКА ПРОВЕДЕННЯ ІНТЕГРОВАНИХ УРОКІВ ТЕХНОЛОГІЙ З ВИКОРИСТАННЯМ ARDUINO	39
3.1 Основні переваги інтегрованих уроків у профільній школі.....	39
3.2 Методика використання Arduino на інтегрованих уроках у профільній школі.....	40
ВИСНОВКИ.....	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	46
ДОДАТКИ.....	49

ВСТУП

Мікроконтролери в даний час широко використовуються для керування технічними пристроями та технологічними процесами. У питаннях вивчення використання мікроконтролерів для управління різними об'єктами можливі два підходи.

Перший підхід - академічний, заснований на глибокому вивченні архітектури мікроконтролерів та технології розробки додатків на їх основі. Цей підхід відрізняється, з одного боку, ефективнішим використанням ресурсів мікроконтролерів у цільовому рішенні, але, з іншого боку, великим інтервалом часу, необхідним для отримання результату.

Другий шлях освоєння мікропроцесорної техніки – це підхід практиків і любителів, заснований на копіюванні готових розробок і не передбачає глибокого проникнення в архітектуру мікроконтролерів. Цей підхід дає змогу отримати швидкий результат для нескладних проектів. Однак для складних пристроїв такі рішення можуть виявитися неефективними і потребують тривалого часу для налагодження та налаштування.[12]

Беручи до уваги те, що цифрові технології не тільки активно застосовуються в наукових дослідженнях але й входять у наше життя: «розумний будинок», автоматичне та дистанційне управління електричними системами, робототехніка. Саме поява мікропроцесорних систем з відкритим вихідним кодом таких як Arduino, заснованих на простому у використанні апаратному та програмному забезпеченні, створила сприятливі умови для їх активного використання в освіті [6].

Особливість таких систем, з точки зору використання їх в освітньому процесі полягає в можливості об'єднати конструювання та програмування в одному уроці, що сприяє інтегруванню викладання природничих наук з розвитком інженерного мислення через технічну творчість.

Водночас вони можуть розглядатися і як засіб підтримки навчального процесу, або, більш ширше – як ефективний засіб цифровізації освітнього середовища.

Це зумовило вибір *теми* кваліфікаційної роботи: «**Розробка й виготовлення моделі Smart Home на базі Arduino та методика використання при вивченні технології у профільній школі.**»

Мета роботи: розробити модель Smart Home на базі Arduino та методику використання при вивченні технології у профільній школі.

Об'єкт: процес розробки й виготовлення моделі Smart Home на базі Arduino.

Предмет: особливості застосування програмно-апаратної платформи Arduino при вивченні технології у профільній школі.

У відповідності до мети визначено такі **завдання:**

1. Дослідити особливості застосування програмно-апаратної платформи Arduino в навчальному процесі.
2. Розробити ескізний проєкт Smart Home на базі Arduino.
3. Виготовити модель Smart Home на базі Arduino.
4. Описати методику проведення інтегрованих уроків технологій з використанням Arduino

У процесі виконання кваліфікаційної роботи були застосовані такі **методи:** аналіз, порівняння та узагальнення вітчизняного й світового досвіду, вивчення технічної документації, проектування, моделювання.

Практична значущість роботи полягає в тому, що створена модель Smart Home на базі Arduino може бути використана в практиці навчання учнів при вивченні технології у профільній школі.

1. ЗАСТОСУВАННЯ ПРОГРАМНО-АПАРАТНОЇ ПЛАТФОРМИ ARDUINO В ОСВІТНЬОМУ ПРОЦЕСІ

1.1 Застосування програмно-апаратної платформи Arduino в освітньому процесі

В даний час Arduino – одна з найзручніших платформ для розробки пристроїв керування на мікроконтролерах. Плата Arduino містить: мікроконтролер ATmega фірми Atmel, схеми скидання, кварцовий резонатор, вбудований стабілізатор напруги живлення, USB-адаптер, що забезпечує зв'язок з ПК, вбудований програматор, засоби внутрішньосхемного програмування [12]. Так званий поріг входження до апаратно-програмних засобів на основі Arduino нижче, ніж для платформи на основі AVR-Studio, де для розробок необхідні знання архітектури мікроконтролерів та мов програмування C++ та/або Ассемблера. У Arduino мова програмування ґрунтується на C/C++, але має спрощений синтаксис (правила граматики в мовах програмування) і відносно проста в освоєнні [6].

Платформа Arduino забезпечує відносно просту розробку додатків на основі AVR-мікроконтролерів та має ряд переваг перед іншими платформами з точки зору навчання та освоєння технології розробки мікроконтролерних пристроїв:

- Низька вартість. Плати Arduino відносно дешеві в порівнянні з іншими платформами;
- Крос-платформність. Програмне забезпечення Arduino працює під ОС Windows, Macintosh OSX та Linux;
- Просте та зрозуміле середовище програмування. Середовище Arduino підходить як для користувачів-початківців, так і для досвідчених;

- Програмне забезпечення з можливістю розширення та відкритим вихідним текстом випускається як інструмент, який може бути доповнений досвідченими користувачами. Мова може доповнюватись бібліотеками C++.

Модулі Arduino являють собою апаратні засоби з можливістю розширення та відкритими принциповими схемами. Схеми модулів випускаються з ліцензією Creative Commons [14], отже, досвідчені інженери мають можливість створення власних версій модулів, розширюючи і доповнюючи їх.

Вказані вище переваги є визначальними при виборі платформи для використання в навчальному процесі розробок на основі мікроконтролерів.

Необхідно вказати ще одну особливість платформи Arduino з точки зору використання в навчальному процесі. Це відносна дешевизна порівняно з промисловим лабораторним обладнанням. Так, наприклад, лабораторна установка на основі NI MyRIO, що випускається National Instruments, спирається на програмне середовище Lab View. Вартість NI MyRIO з програмним середовищем Lab View, залежно від складу модулів, може сягати кількох сотень тисяч гривень. Ціна ж одного лабораторного комплексу, що включає два Arduino Uno, макетну плату зі сполучними провідниками, елементами індикації, що керують перемикачами, обійдеться менше ніж п'ять тисяч гривень.

Таким чином, платформу Arduino можна використовувати в навчальному процесі для створення проєктів з дослідження мікроконтролерів та технології розробки додатків, що відрізняються високою функціональністю, зручним середовищем програмування та низькою вартістю, зумовленою відкритістю схем та програмного забезпечення.

Навчання учнів на уроках технології з використанням таких лабораторних установок дозволить скоротити час освоєння ними технологій розробки мікроконтролерів керуючих пристроїв.

1.2 Можливості програмно-апаратної платформи Arduino в освітньому процесі

Під брендом Arduino виробляється ціла лінійка апаратно-програмних комплексів, які використовуються для автоматизації при побудові та прототипуванні різних технічних систем, так чи інакше пов'язаних з електронікою. Основою апаратної частини Arduino є друкована плата з мікроконтролером та мінімальною електронною обв'язкою до нього. Функціонал основної плати може доповнюватися картами розширення (шилдами), які забезпечують істотне збільшення можливостей.[12]

Найважливішою особливістю даної платформи є повністю відкрита архітектура (як програмна, так і апаратна), що дозволяє вільно копіювати і доповнювати продукцію офіційного розробника. В результаті безліч сторонніх виробників випускає широке коло обладнання (датчиків, виконавчих пристроїв та ін), сумісного з Arduino, що призводить до додаткового зростання її популярності.

Таким чином, Arduino може виступати як зручна платформа швидкої розробки електронних пристроїв. Важливо, що ця платформа також відрізняється простотою і не пред'являє високі вимоги до розробника, що дозволяє використовувати її під час навчання в школі. Як недолік відзначається менша надійність платформи в порівнянні з промисловими мікроконтролерами, однак цей недолік малозначимий, якщо мова йде про використання Arduino в сфері освіти. На даний момент існує багато навчальної літератури [15; 10; 20], наукових та методичних статей, у яких

обговорюються переваги та недоліки цієї платформи, а також її використання в освітньому процесі [6; 12; 13; 16].

З педагогічної точки зору використання наборів Arduino має ряд важливих переваг:

1. Стимулювання мотивації учнів до отримання знань;
2. Розвиток інтересу учнів до техніки, програмування та конструювання;
3. Формування навичок програмування, розвитку логічного і алгоритмічного мислення [12].

Все це повною мірою відноситься до проєктів створених на основі платформи Arduino. Використання платформи Arduino в педагогічній діяльності відкриває нові можливості для школяра. Проєкти, що реалізуються в навчальному процесі в загальноосвітніх установах, можуть виконуватися і розвиватися вдома. Крім розкриття творчих здібностей створюється благодатна основа для реалізації індивідуалізації навчального процесу.

Враховуючи відносно дешевизну плат Arduino, датчиків і необхідних для розробки елементів, а також зацікавленість учнів у володінні власним комплектом Arduino, можлива дистанційна форма навчання роботі з даною мікропроцесорною системою.

Для успішної фронтальної роботи в режимі відеоконференції учням необхідно мати однаковий, заздалегідь узгоджений комплект елементів і покроковий набір інструкцій. Вчитель, таким чином, не тільки може демонструвати збірку, підключення та роботу пристрою, але і може допомогти у вирішенні проблемної ситуації.

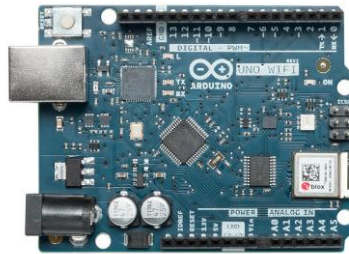
Таким чином, застосування сучасних технологій в освітньому середовищі дозволяє формувати в учнів не тільки запас глибоких наукових знань, а й розвивати в них певні навички в галузі техніки та ІТ-технологій. Використання платформи Arduino на уроках технологій в профільній школі

дозволяє відкрити нові можливості для школярів, сприяє розкриттю творчих здібностей, індивідуалізації навчального процесу, створенню спеціальних умов для учнів з обмеженими можливостями.

2. РОЗРОБКА Й ВИГОТОВЛЕННЯ МОДЕЛІ SMART HOME НА БАЗІ ARDUINO

2.1 Особливості програмно-апаратної платформи Arduino

Arduino – це повністю відкрита програмно-апаратна платформа для створення систем роботизації та автоматизації (у тому числі домашньої) різної складності. Її основна відмінність – простота у використанні, можливість освоєння та створення цілком працездатних проектів навіть без глибоких знань електроніки та програмування.[20]



Апаратні засоби Arduino дозволяють отримати сигнали від сенсорів (наприклад, рівень температури, наявність світла, натискання на кнопку) або складніших систем, таких як html-сторінки.

Програмний код обробляє їх і видає сигнали на виконавчі механізми, наприклад включає сервопривід, замикає контакти реле або сервіси, здатні виконувати дуже складні операції – від відправки повідомлень на e-mail до розміщення контенту на сторінках.

Основними перевагами платформи є:

Повна відкритість. Це стосується як апаратних модулів (наводяться схеми й у більшості – друковані плати), і програмного забезпечення. В результаті сторонні розробники мають можливість створювати повністю сумісні пристрої. Спільноту створено безліч програмних проектів та модулів для вирішення різних завдань автоматизації. Всі вони знаходяться у вільному доступі, так що підібрати необхідний софт і адаптувати його під власні потреби не складно.

Широкий вибір. В асортименті пропонованих продуктів – рішення для будь-якого рівня, від плат на 8-бітових контролерах з мінімумом функціональних можливостей для початкових кроків та простих проектів до потужних контролерів на 32-бітових чіпах. Можливості підбору необхідної конфігурації розширює багатий асортимент своїх модулів розширення та апаратних продуктів сторонніх розробників. Крім того, придбати продукти Arduino можна практично у будь-якому вигляді – від друкованої плати з набором компонентів для збирання своїми руками, до готових контролерів та Starter Kits – спеціалізованих наборів.

Кросплатформність. Софт Arduino працює під усіма ОС для ПК – Windows, Linux, MacOS, зберігаючи повну функціональність, а готові програмні напрацювання взагалі є платформно незалежними.

Гнучкість підходу. На платах Arduino реалізований весь необхідний функціонал – від стабілізатора живлення та програматора для контролера, до набору портів вводу/виводу та конкретних інтерфейсів, наприклад Bluetooth, Wi-Fi, управління двигунами та ін. Однак представлена система не обмежує розробника у виборі, наприклад, прошивку мікроконтролера можна виконати не лише за допомогою штатних засобів, але також і за допомогою сумісності програматорів інших виробників, а набір портів та інтерфейсів легко розширити.

Простота освоєння. Прозорі схемотехнічні рішення, якісна документація дозволяють працювати з модулями платформ без серйозного досвіду та знань електроніки. Цьому сприяє програмне середовище, що вимагає лише базових знань C++. Більш того, проект пропонує і засоби графічного проектування, освоїти які можуть навіть діти та підлітки.

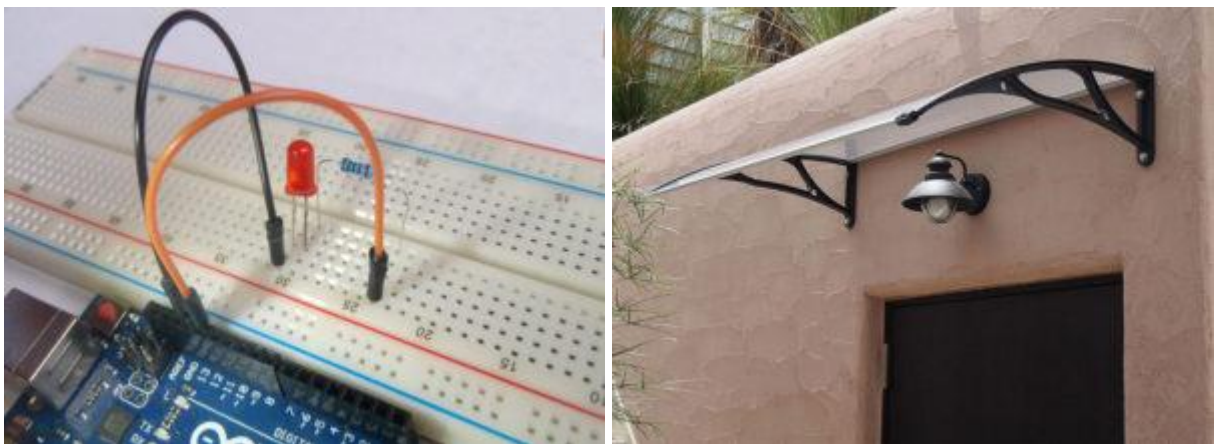
Вартість. Мікроконтролери Arduino на сьогоднішній день є одним з найбільш доступних у ціновому середовищі варіантів серед множинних модифікованих аналогів – навіть найпотужніший мікроконтролер

обійдеться покупцеві не дорожче 50 доларів. На ринку можна знайти безліч різноманітних розширень і компонентів, що випускаються різними виробниками, у тому числі й азіатськими, які за досить гідної якості виготовлення коштують недорого.[6]

2.2 Етапи розробки ескізного проєкту Smart Home на базі Arduino

Створення розумного будинку на Arduino своїми руками мало чим відрізняється від розробки та реалізації будь-якої іншої електронної системи. Процес включає кілька обов'язкових етапів. Відповідно, розробнику доведеться вирішити кілька задач.

Формулювання завдання.



Завдання для проєкту має бути визначене максимально точно та докладно. Наприклад, вирішується одна з приватних home automation – увімкнути освітлення ганку, коли у темний час до нього наближається людина. Цей загальний опис завдання потребує більш конкретних формулювань для проєктування.

Вони описують:

Основні умови для спрацювання автоматики – тригери (triggers) у термінах home automation. У розглянутій задачі такий тригер один – наближення людини до ганку.

Додаткові умови (conditions). Вони визначають дозвіл на спрацювання тригера і, як правило, визначають тривалість їх дії. Для цього завдання додаткова умова – темний час. Справді, вмикати світло на ганку вдень не має сенсу. Немає необхідності робити це і коли ганок освітлений – світло вже включене або задіяні інші джерела. Відповідно, додаткові умови можна задати в декількох варіантах: настав темний час доби, і світло на ганку не ввімкнено. Практично повний опис умов, але не враховує можливість освітлення з інших джерел, освітленість на ганку нижче порогового значення. Хоча умова виглядає простіше, але враховує практично всі випадки.

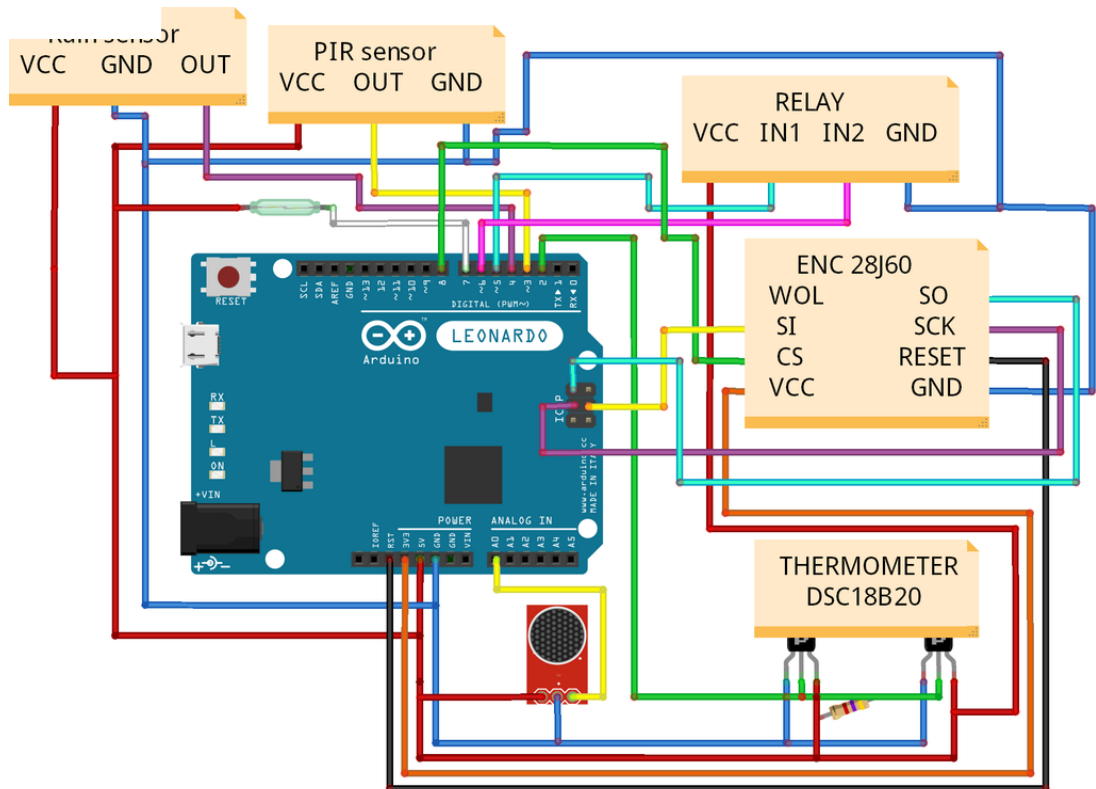
Дії (Actions). Описує реакцію системи на тригери та додаткові умови. Для поточного прикладу система має виконати єдину дію – увімкнути світло.

Дії у разі відсутності тригерів та/або невиконання умов. Тут необхідно розглядати два випадки: ні основні, ні додаткові умови не були виконані. Найбільш логічна поведінка – бездіяльність, тригер та умови були виконані, відбулося спрацювання, потім ситуація змінилася (т.зв. post-action дії). Поведінка проектованої системи має виглядати так – не включати світло до спрацювання тригера і виконання умов, вимкнути після того, як він перестав бути потрібен.

Таким чином, результат – докладний опис завдання:

- Перевіряти, чи має ганок людина.
- Не діяти, поки його немає.
- Якщо людина підійшла – перевірити освітленість.
- Якщо вона нижче заданої – увімкнути світло.
- Перевіряти, чи залишається потрібне світло.
- Якщо так – залишатись у поточному стані, якщо ні – вимкнути світло.

Вибір способів реалізації алгоритму.



На цьому етапі визначається набір сенсорів для отримання даних (тригерів та умов) та виконавчих механізмів для виконання дій.

Навіть у найпростішому даному випадку можливі кілька варіантів реалізації. Наприклад, визначати присутність людини біля ганку можна:

- за сигналом стандартного інфрачервоного датчика;
- розмістивши під майданчиком біля ганку датчики, що реагують на тиск (наприклад, тензодатчики на основі п'єзофекту);
- отримуючи з Wi-Fi-роутера, що працює в квартирі, сигнал про підключення нового/конкретного абонента до домашньої бездротової мережі і т.д.

Аналогічним чином розглядаються ситуації з визначенням рівня освітленості та включенням світла на ганку.

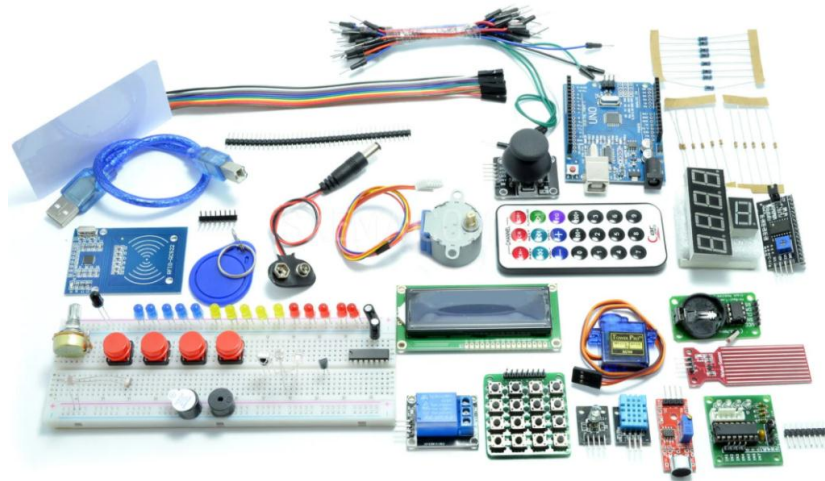
Завершує етап запис алгоритму з урахуванням прийнятих рішень, наприклад:

- Опитувати провідний ІЧ-датчик.
- Якщо сигналу немає, нічого не робити.
- При появі сигналу перевірити стан провідного датчика освітленості.
- За наявності сигналу нічого не робити.
- За відсутності сигналу – видати на ZigBee хаб команду на включення вимикача за відповідною адресою (що відповідає за освітлення на ганку).
- Опитувати ІЧ датчик.
- Поки що є сигнал (людина на ганку) – нічого не робити.
- Якщо сигналу немає – вимкнути світло, тобто. видати на ZigBee вимикач команду відключення.
- Повернутися на початок циклу.

Ще один результат цього етапу – перелік необхідного обладнання та матеріалів. Для складання системи знадобляться:

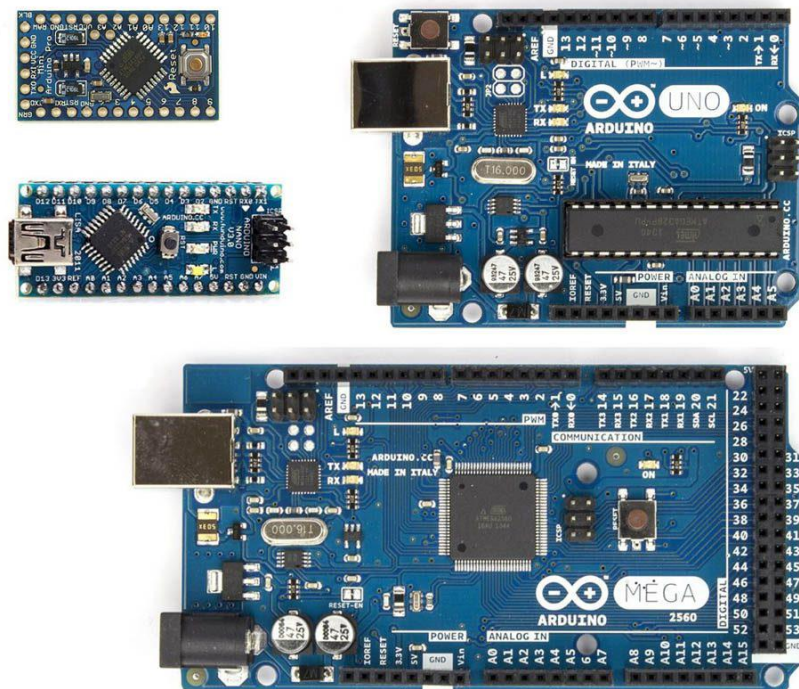
- Контролер із можливістю опитування провідних каналів зв'язку (мінімум, двох, для датчика освітленості та ІЧ-датчика).
- ZigBee вимикач.
- ZigBee хаб (без нього моно обійтися, якщо контролер має інтерфейс прямого управління ZigBee пристроями).

Вибір обладнання.



Після складання структурної схеми проекту можна розпочинати розробку принципової, яка, передусім, включає вибір устаткування.

Вибір контролера Arduino.



Оскільки базова платформа Arduino, визначена раніше, залишається вибрати конкретну модель контролера і, при необхідності, розширення. Завдання не складне, оскільки асортимент пристроїв досить великий і включає моделі для більшості практичних завдань[24].

До списку обладнання входять:

Контролер	Чип	Пам'ять (ОЗП/ПЗП)	Порти вводу/виводу та інтерфейси	Додатково
Due	32-бітний Cortex-M3 ARM SAM3U4E, 84 МГц	64+36Кб/256+256 Кб (можлива пряма адресація всього простору)	54 цифрових ІО (деякі із спеціальними функціями, наприклад, ШІМ) 12 аналогових входів (з АЦП) 2 аналогових входа	Живлення контролера 3.3В
Uno	8-біт АТmega328 або АТmega8U2 (у новій версії), 16 МГц	2Кб/32Кб+1Кб EEPROM	14 цифрових ІО (6 можуть використовуватися як ШІМ) 6 аналогових входів з АЦП UART TTL USB Роз'єми для модулів розширення	
Leonardo	8-біт АТmega32u4, 16МГц	2.5кб/32Кб+1Кб EEPROM	20 цифрових ІО (7 з функціями ШІМ) 12 аналогових входів (можуть працювати як цифрові) USB розширення	Вбудована підтримка USB дає змогу налаштовувати видимість при зв'язку з ПК (як клавіатуру, мишу, віртуальний послідовний порт) Функціонально та конструктивно аналогічний Uno
Mini	8-біт АТmega168, 16МГц	1Кб/16Кб+512б EEPROM	14 цифрових ІО (6 можуть використовуватися як ШІМ) 6 аналогових входів з АЦП	Для програмування потрібен адаптер MiniUSB Компактне рішення для макетування
Micro	8-біт АТmega32u4, 16МГц	Компактний контролер, за функціональністю		

		та аналогічним Leonardo, для підключення та програмування використовується кабель microUSB		
Nano	8-біт ATmega168 або ATmega328, 16МГц	Компактний варіант Duemilanove, що відрізняється лише габаритами та відсутністю роз'єму для підключення зовнішнього джерела живлення		
Mega	8-біт ATmega1280, 16МГц	8Кб/128Кб+4Кб EEPROM	54 цифрових ІО (14 з функціями ШІМ) 16 аналогових входів з АЦП 4 порти UART USB розширення	
Pro	8-біт ATmega328, 16МГц (3.3В) або ATmega168, 8МГц (5В)	2 або 1 Кб/32 або 16 Кб + 1 Кб або 512б EEPROM	2 або 1 Кб/32 або 16 Кб + 1 Кб або 512б EEPROM 14 цифрових ІО (6 можуть використовуватися як ШІМ) 6 аналогових входів з АЦП UART TTL USB Роз'єми для модулів розширення	Роз'єми введення/виведення та інтерфейсів не встановлені, розробник може використовувати власні

При виборі контролера необхідно враховувати:

- Продуктивність;
- Наявність необхідної кількості портів введення/виводу, цифрових та аналогових;

- Об'єм пам'яті для зберігання програм і даних (усі плати Arduino використовують небагаті ресурси мікроконтролера і передбачають розширення);
- Зручність роботи – програмування, налагодження програм, встановлення готового пристрою на корпус.

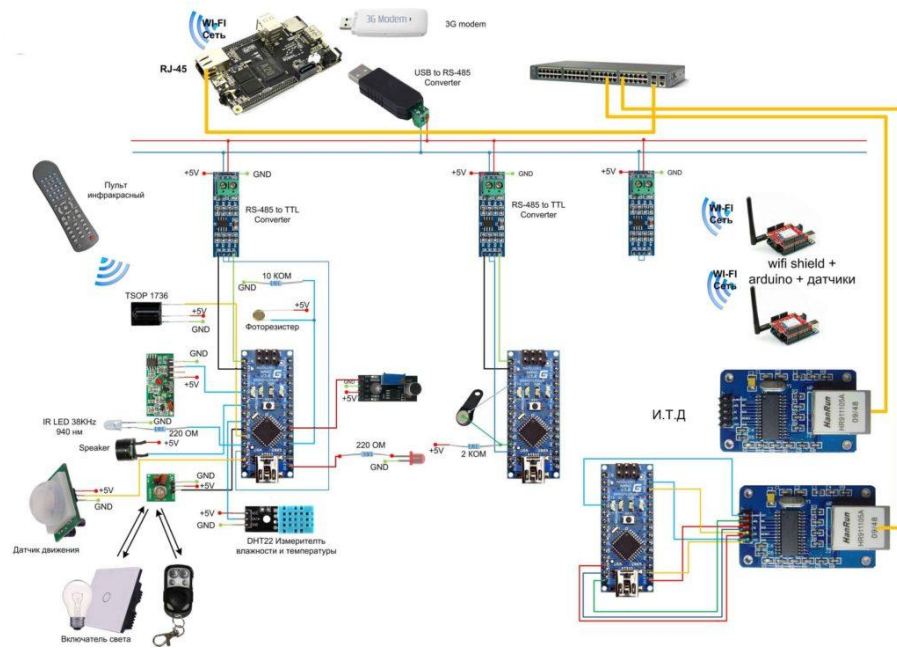
Вибір периферії.



При виборі датчиків та виконавчих механізмів немає значних обмежень. Єдина вимога – забезпечити сумісність із портами Arduino за рівнями сигналів та навантаження.

Однак, за бажання, і ця вимога легко обходиться за рахунок складання власних чи використання готових плат узгодження.

Розробка принципової схеми.



Після вибору контролера та периферійних пристроїв слід намалювати схему системи автоматизації. З платами Arduino це не складно, головне завдання розробника – вибрати піни портів введення/виводу для підключення.

Робота з програмним забезпеченням.

Коли схема готова, починається найважливіший етап проектування – написання програми для контролера. Для цього потрібно завантажити середовище розробки Arduino IDE.

Програмне забезпечення викачують зі сторінки офіційного сайту <https://www.arduino.cc/en/software>.

Стабільна версія - Arduino IDE 1.8.13 підтримує:

- Windows 7 і новіші версії ОС, x86 і x64;
- MacOS 10.10 і більше;
- 32- та 64-бітові версії Linux та Linux ARM.

Для роботи під Windows можна завантажити інсталятор або ZIP-архів софту. Використовувати інсталятор для недосвідчених користувачів краще – установка драйверів буде зроблено автоматично разом з інсталяцією програми.

При використанні ZIP-архіву достатньо розгорнути програму на жорсткому диску ПК та запустити IDE. У цьому випадку програма також спробує встановити драйвера, але можливо, доведеться оновити їх вручну в Панелі керування. Усі необхідні .inf файли входять до комплекту постачання.

Можливості середовища розробки Arduino IDE:

- Створення проектів (скетчою, від англ. sketch - малюнок).
- Їх перевірка та компіляція для завантаження в контролер;
- «Заливання» готового машинного коду в контролери;
- Підтримка всіх версій плат;
- Підключення бібліотек, які не входять до комплекту постачання.

Покроково процес виглядає так:

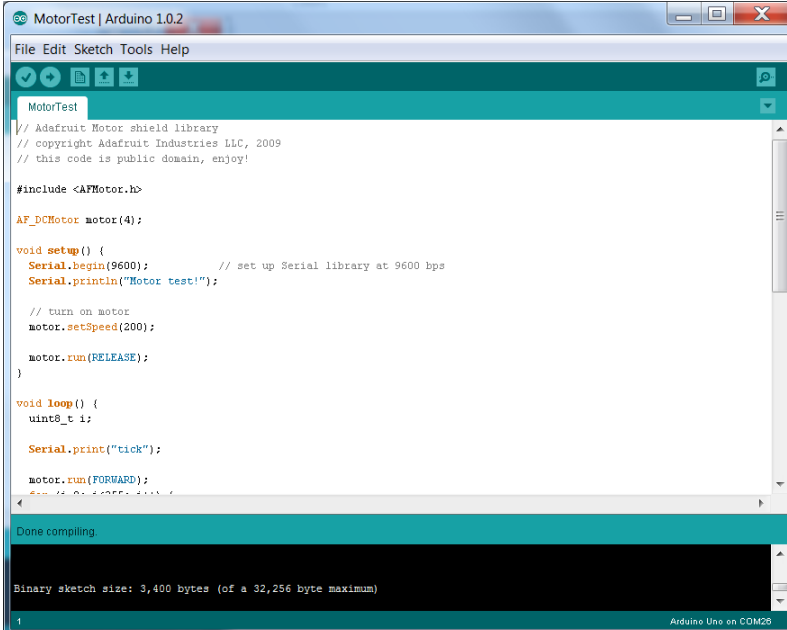
1. Підключити плату Arduino кабелем до порту USB комп'ютера.
2. Запустити Arduino IDE.
3. У розробці відкрити готовий скетч – у меню File вибрати папку Examples-1.Basic, у ній файл Blink.
4. Вибрати плату контролера до роботи – меню Tools-Board.
5. Вказати порт для зв'язку з платою (як правило COM3 або інший COM з більшим номером). Якщо цих портів немає, слід

перевірити, чи підключена плата або встановити драйвери вручну.

6. Відкомпілювати скетч через меню Sketch-Verify/Compile або натисканням на відповідну кнопку панелі інструментів під рядком меню.
7. Завантажити готовий скетч до контролера (Sketch-Upload) або натисканням кнопки панелі інструментів. На цьому етапі можна спостерігати миготіння світлодіодів Rx та Tx, яке свідчить про обмін з ПК.

Після завантаження через кілька секунд почне блимати світлодіод на платі Uno. Це свідчить, що і контролер і середовище розробки повністю працездатні.

Програмування скетчів



```

MotorTest | Arduino 1.0.2
File Edit Sketch Tools Help
MotorTest
// Adafruit Motor shield library
// copyright Adafruit Industries LLC, 2009
// this code is public domain, enjoy!

#include <AFMotor.h>

AF_DCMotor motor(4);

void setup() {
  Serial.begin(9600); // set up Serial library at 9600 bps
  Serial.println("Motor test!");

  // turn on motor
  motor.setSpeed(200);

  motor.run(RELEASE);
}

void loop() {
  uint8_t i;

  Serial.print("tick");

  motor.run(FORWARD);
}

Done compiling

Binary sketch size: 3,400 bytes (of a 32,256 byte maximum)
  
```

Програмування скетчу можна виконати безпосередньо в редакторі середовища розробки або будь-якому іншому зручному текстовому редакторі (в останньому випадку не забути зберегти роботу у файлі .ino). Вбудована мова програмування Arduino є клоном мови C++ з деякими спрощеннями та

додатковими функціями та бібліотеками для звернення до функціонала контролера.[21]

Для роботи з програмою в IDE необхідно:

1. Створити новий скетч (File-New).
2. Підключити необхідні бібліотеки. Стандартні бібліотеки функцій входять до комплекту поставки, нестандартні потрібно розмістити у відповідних папках Libraries. Після цього їх імена стануть доступними розробнику у меню Sketch-Import Library. Достатньо вибрати потрібні імена, і вони будуть увімкнені для обробки препроцесором.
3. У функції `setup()` прописати всі дії для ініціалізації системи. Ця функція виконується один раз перед початком. Як правило, у її тексті достатньо обмежитися призначенням функцій пін введення/виводу.
4. У функції `loop()` прописати код, який реалізує алгоритм автоматизації – опитування сенсорів, аналіз умов, видачу сигналів на виконавчі механізми. Функція виконується у нескінченному циклі, оператори переривання цього циклу не передбачені.
5. Відкомпілювати написаний скетч.
6. Завантажити його у контролер.

З усіма платами контролерів та розширень поставляються необхідні бібліотеки для роботи з усім апаратним забезпеченням. Крім того, в мережі доступні тисячі готових бібліотек, розроблених спільнотою. Підібрати потрібну (з документацією) можна за посиланнями на офіційному сайті <https://playground.arduino.cc/Main/LibraryList/> та <https://www.arduinolibraries.info/> або GitHub.

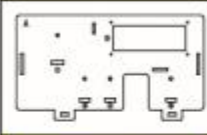





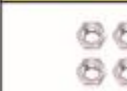




2.3 Розробка й виготовлення моделі Smart Home на базі Arduino

Основою для побудови Smart Home є панелі виготовлені з листової фанери:

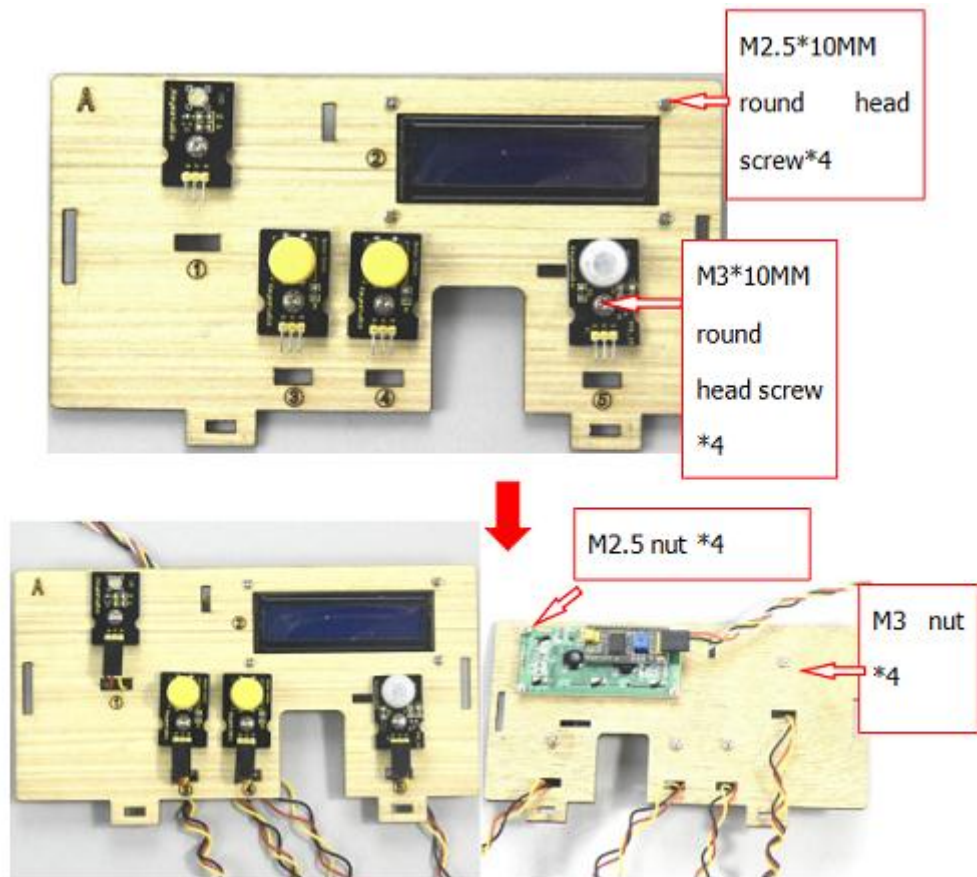


Крок 1: Встановлення датчиків плати А

Необхідно підготувати А board*1, М3*10ММ round screw*4, М3 nickel plated nut*4 ; М2.5*10ММ round screw*4, button sensor*2, white LED*1, PIR motion sensor*1, LCD1602 display*1, 4pin F-F dupont line*1, 3pin F-F dupont line*4

A Board*1	Button module*2	White LED*1	PIR motion sensor*1	LCD1602 Display*1	4pin F-F Dupont line*1
					
M2.5 Nickel plated nut*4	M3 Nickel plated nut*4	M2.5*10MM Round head screw*4	M3*10MM Round head screw*4	3pin F-F Dupont line*4	
					

Для цього потрібно закріпити білий світлодіод, датчики 2 кнопок і датчик руху PIR на відповідній ділянці плати А за допомогою 4 гвинтів М3*10 мм із круглою головкою та 4 гайок М3. Потім встановити дисплей LCD1602 на плату за допомогою 4 гвинтів М2,5*10 мм із круглою головкою та 4 гайок М2,5. Далі з'єднати їх 3- та 4-контактними лініями Dupont.

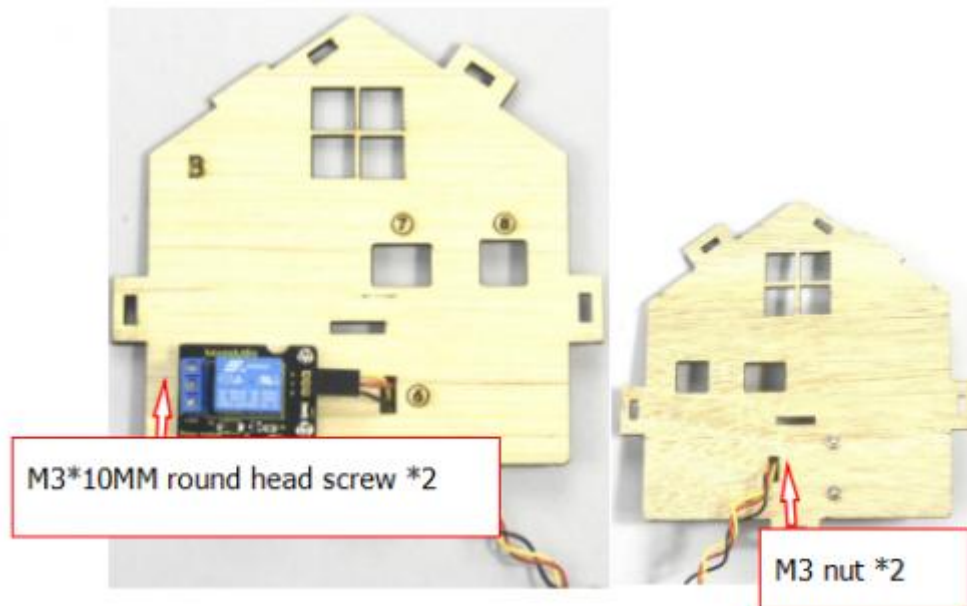


Крок 2: Встановлення датчиків плати В

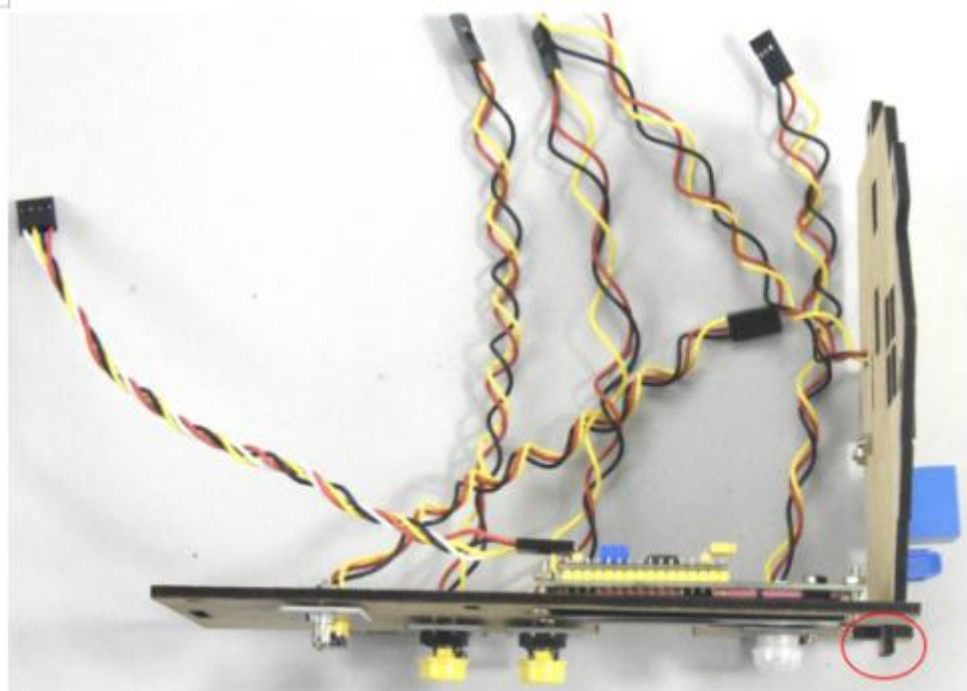
Необхідно підготувати В board, а 3pin F-F Dupont line, 2pcs M3*10MM round head screws, 2pcs M3 nickel plated nuts and a relay module

B Board*1	Relay module*1	M3 Nickel plated nut*2	M3*10MM Round head screw*2	3pin F-F Dupont line*1
				

Збираємо модуль реле на платі В за допомогою 2 шт. гвинтів M3*10 мм і 2 шт. гайок M3, з'єднуємо їх за допомогою 3-контактної лінії Dupont.

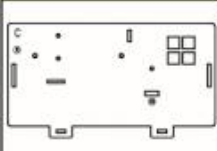








Крок 3: З'єднуємо плати А та В.

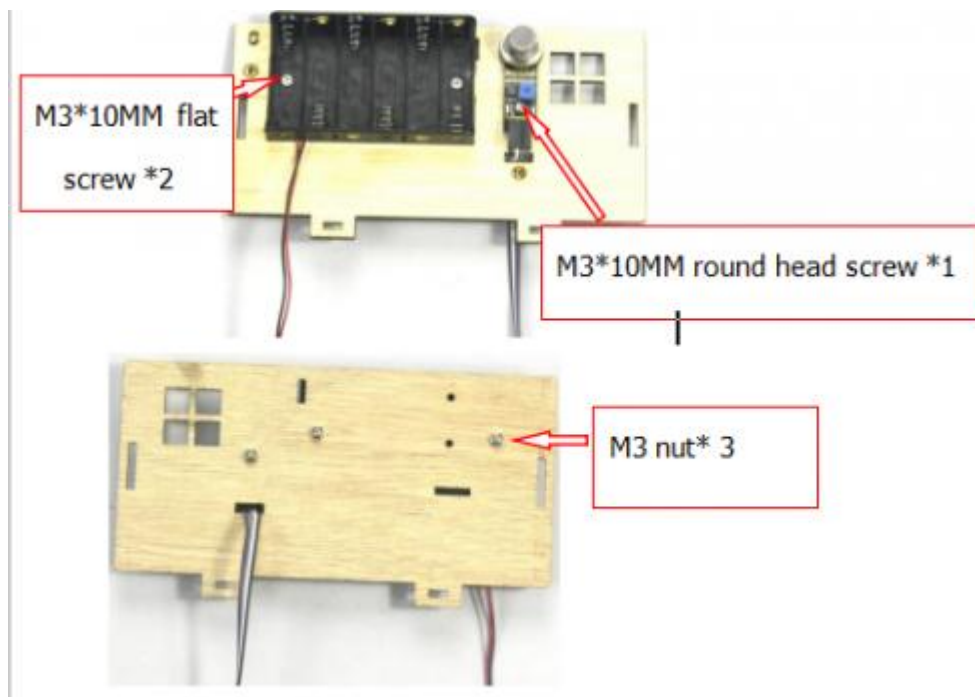


Крок 4: Встановлення датчиків та тримача батареї плати С

Необхідно підготувати С board, MQ-2 gas sensor, battery holder, 2pcs M3*10MM flat head screws, a M3*10MM round head screw, 3pcs M3 nickel plated nuts and 4 F-F dupont lines.

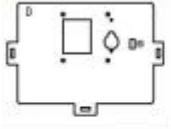










C Board*1	MQ-2 Gas sensor*1	Battery holder*1	M3*10MM Flat head screw*2	M3*10MM Round head screw*1	M3 Nickel plated nut*3	F-F Dupont line*4
						

Для цього потрібно закріпити тримач батареї на платі С за допомогою 2 гвинтів М3*10 мм з плоскою головкою та 2 гайок М2. Потім встановити датчик газу MQ-2 на відповідну ділянку плати С за допомогою гвинта М3*10ММ із круглою головкою та гайки М3. Далі з'єднати їх 4 лініями Dupont.

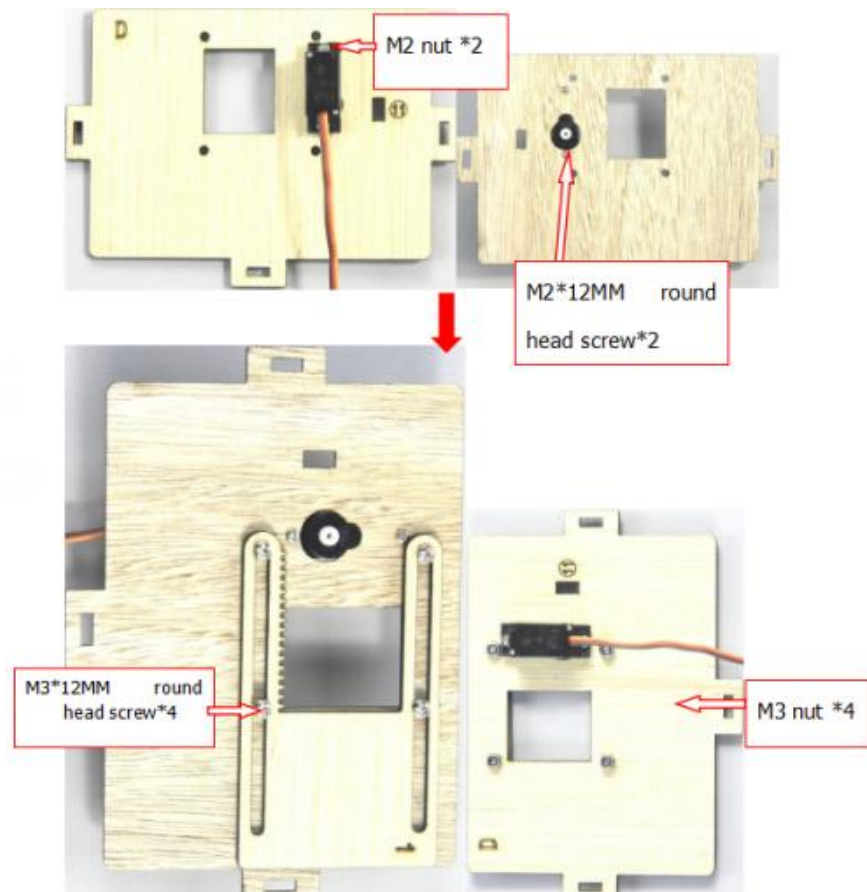


Крок 5: Встановлення датчиків та частин плати D

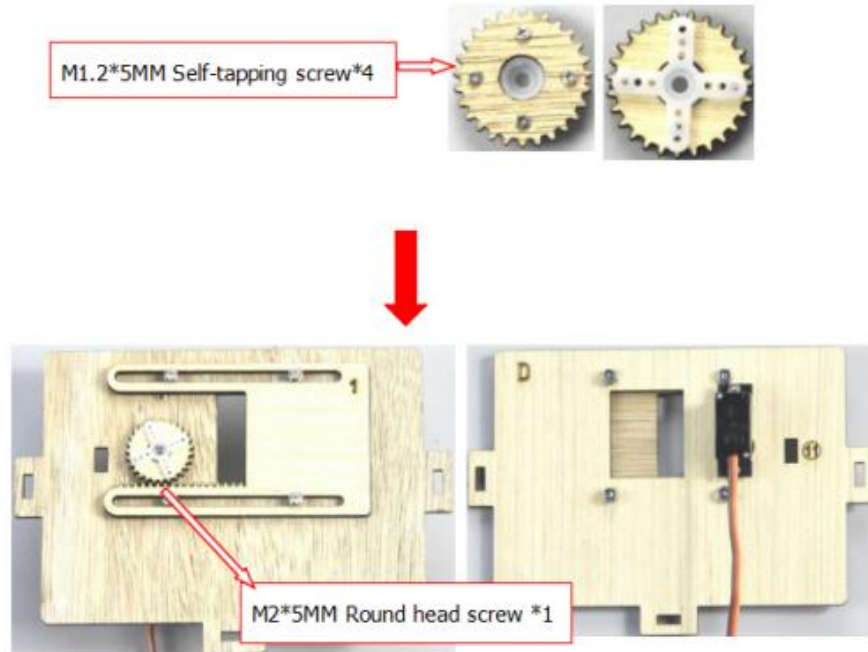
Необхідно підготувати servo, 4pcs M1.2*5 self-tapping screws, a white cross mount (included in servo) , a M2*5 round head screw (included in servo) , 2pcs M2*12MM round head screws, 2pcs M2 nickel plated nuts, 4pcs M3*12MM round head screws, 4pcs M3 stainless self-locking nuts, a D board, a gear, a board1.

D Board*1	Board 1*1	Gear*1	Servo motor*1	White cross mount*1	M2*5 Round head screw*1
					
M2 Nickel plated nut*2	M3 Stainless self-locking nut*4	M3*12MM Round head screw*4	M2*12MM Round head screw*2	M1.2*5 Self-tapping screw*4	
					

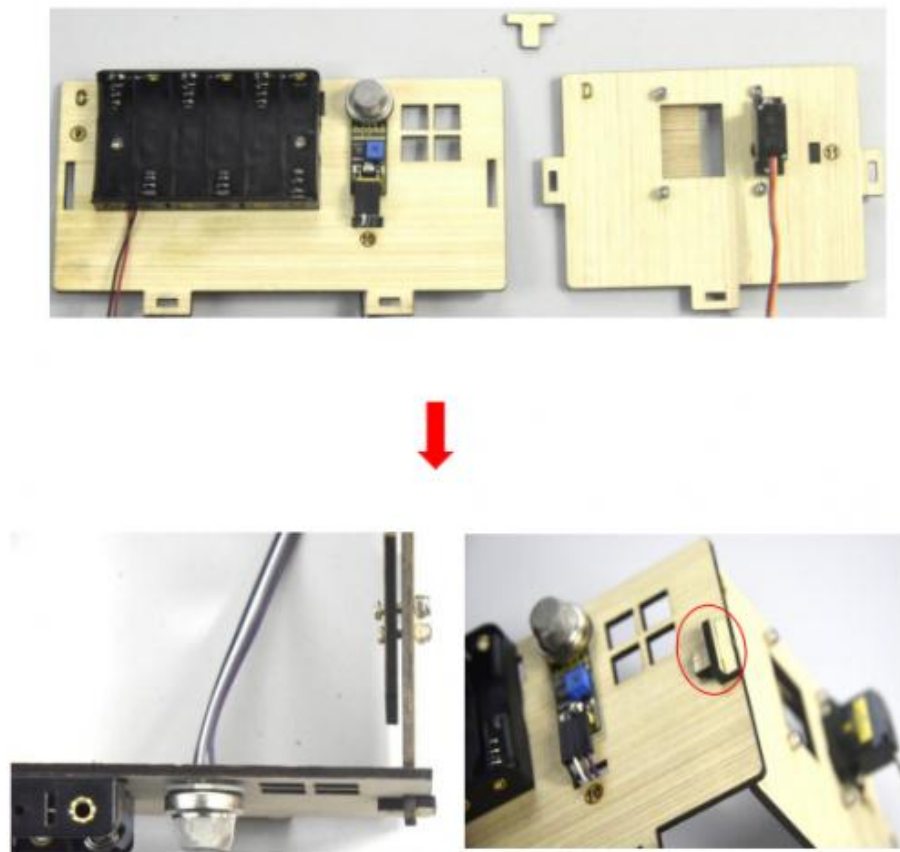
Для цього потрібно закріпити сервопривод у відповідній області на платі D за допомогою 2 гвинтів M2*12 мм із круглою головкою та 2 гайок M2. Потім встановіть квадратну плату 1 на плату D за допомогою 4 гвинтів M3*12 мм із круглою головкою та 4 самоконтряться гайок M3.



Біле хрестоподібне кріплення на шестерні необхідно закріпити за допомогою 4 саморізів M1,2*5 мм і встановити шестерню на серводвигуні за допомогою 1 гвинта з круглою головкою M2*5 мм.



Крок 6: З'єднання плати С із платою D за допомогою болта типу «Т».

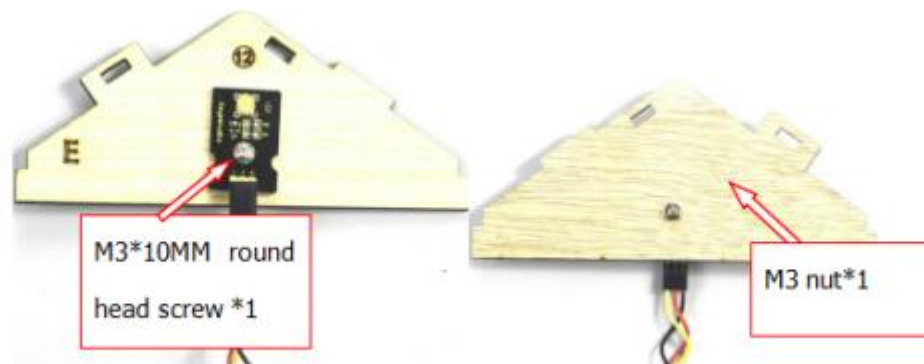


Крок 7: Встановлення датчику плати E.

Необхідно підготувати yellow LED module, a E board, a M3*10MM round head screw, a M3 nickel plated nut and a 3pin F-F Dupont line



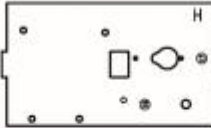
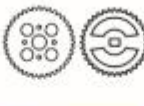
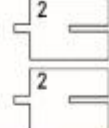











E Board*1	Yellow LED*1	M3 Nickel plated nut*1	M3*10MM Round head screw*1	3pin F-F Dupont line*1
				

Закріплюємо жовтий світлодіод на відповідній ділянці електронної плати за допомогою 1 гвинта M3*10 мм із круглою головкою та 1 нікельованої гайки M3, а потім під'єднуємо 3-контактну лінію Dupont.

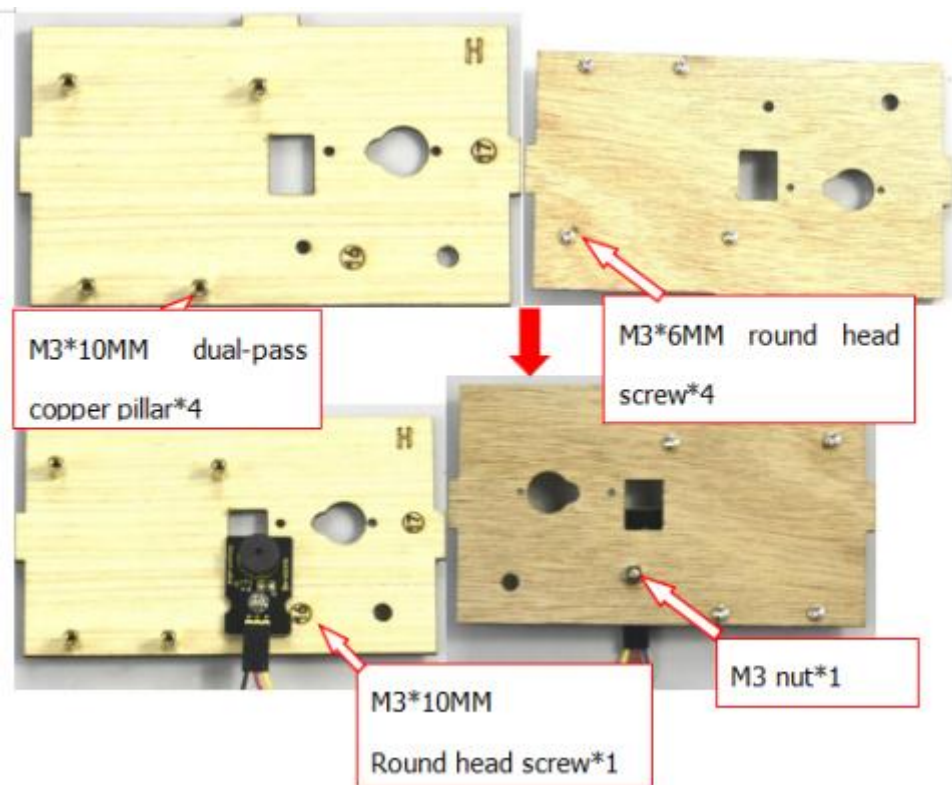


Крок 8: Встановлення плати керування, датчиків та частини плати Н.

Необхідно підготувати servo, a passive buzzer, 4pcs M1.2*5 self-tapping screws, a white cross mount(included in servo), a M2*5 screw(included in servo), 2pcs M2*12MM round head screws, 2pcs M2 nickel plated nuts, a M3*10MM round screw, a M3 nickel plated nut, 8pcs M3*6MM round head screws, 4pcs M3*10MM dual-pass copper pillars, a Keystudio PLUS control board, a sensor shield, a 3pinF-F Dupont line, a board E, 2 gears and 2pcs board 2.

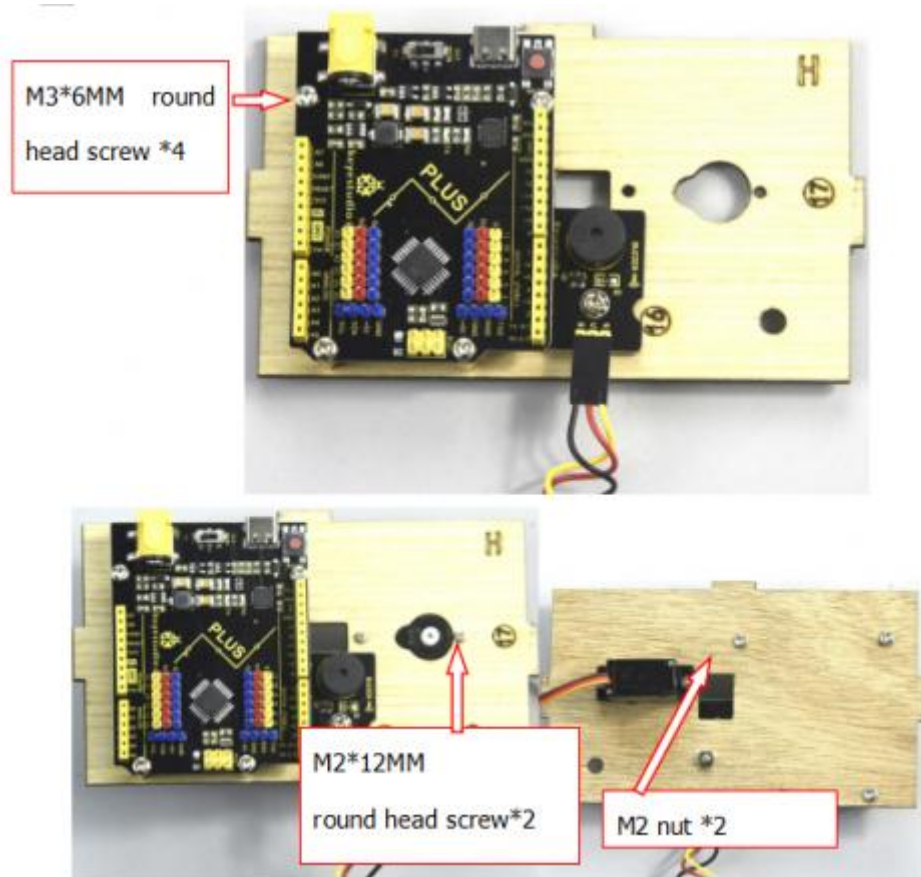
Keyestudio PLUS control board	Sensor shield	H Board*1	Gear*2	Board 2*2	M3*6MM Round head screw*8
					
Servo motor*1	Passive buzzer*1	M2 Nickel plated nut*2	M3 Nickel plated nut*1	M3*10MM Round head screw*1	M2*12MM Round head screw*2
					
M1.2*5 Self-tapping screw*4	White cross mount*1	M2*5 screw*1	M3*10MM Dual-pass copper pillar*4	3pin F-F Dupont line*1	
					

Для цього закріплюємо 4 шт. двохрідних мідних опор на H-платі за допомогою 4 шт. гвинтів M3*6 мм. Потім закріплюємо пасивний зумер на H-платі за допомогою 1 гвинта M3*10MM із круглою головкою та 1 гайки M3. Далі з'єднуємо їх за допомогою 3-контактного дроту Dupont.



Сервопривід перед установкою повертаємо на 90°, таким самим способом, як у кроці 6. Закріплюємо 4 мідні опори M3*10MM на платі керування keyestudio PLUS за допомогою 4 гвинтів M3*6MM із круглою

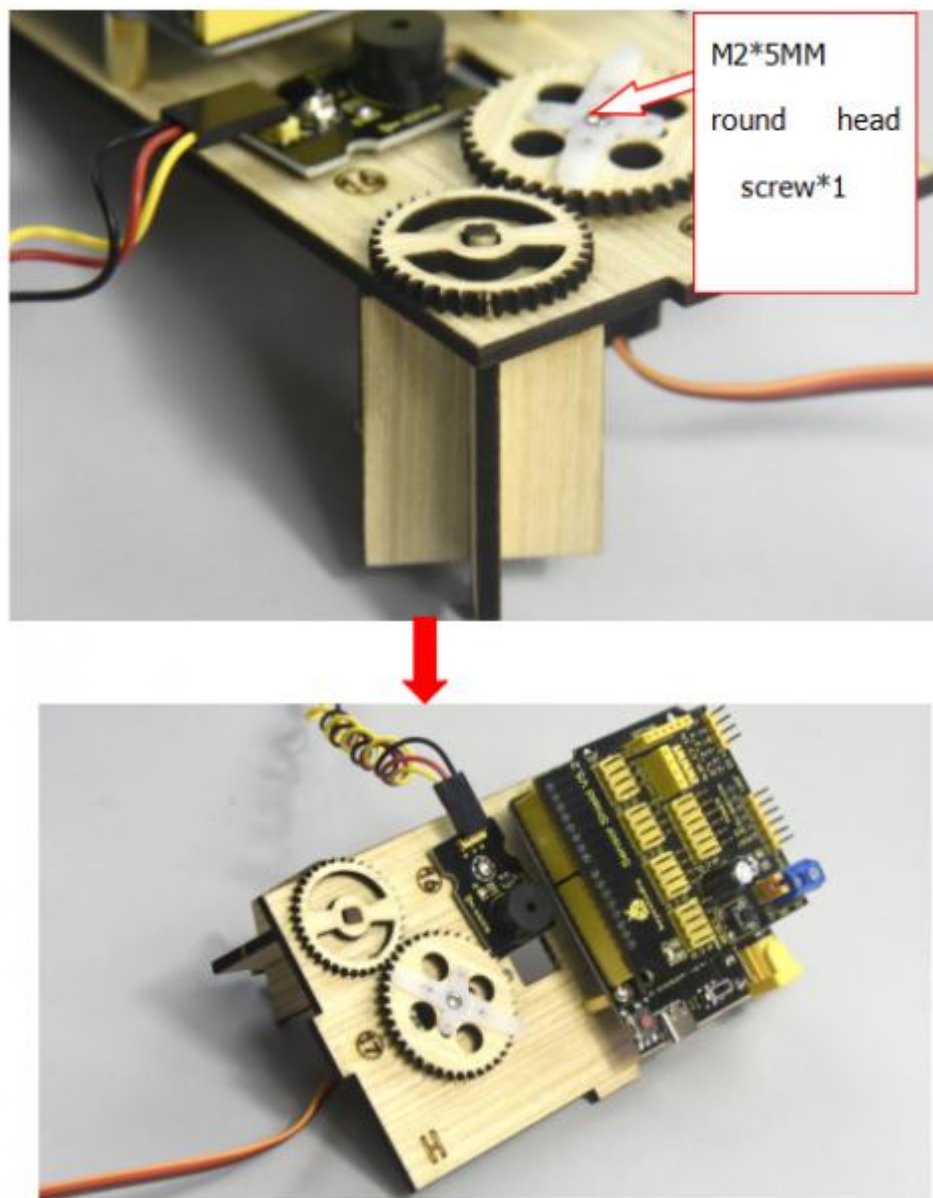
головкою, а потім закріплюємо сервопривід на відповідній ділянці плати Н з 2 гвинтами M2*12 мм із круглою головкою та 2 гайками M2.



Потім необхідно зібрати разом 2 панелі 2, і закріпить білий хрестик на шестерні за допомогою 4 саморізів M1,2*5

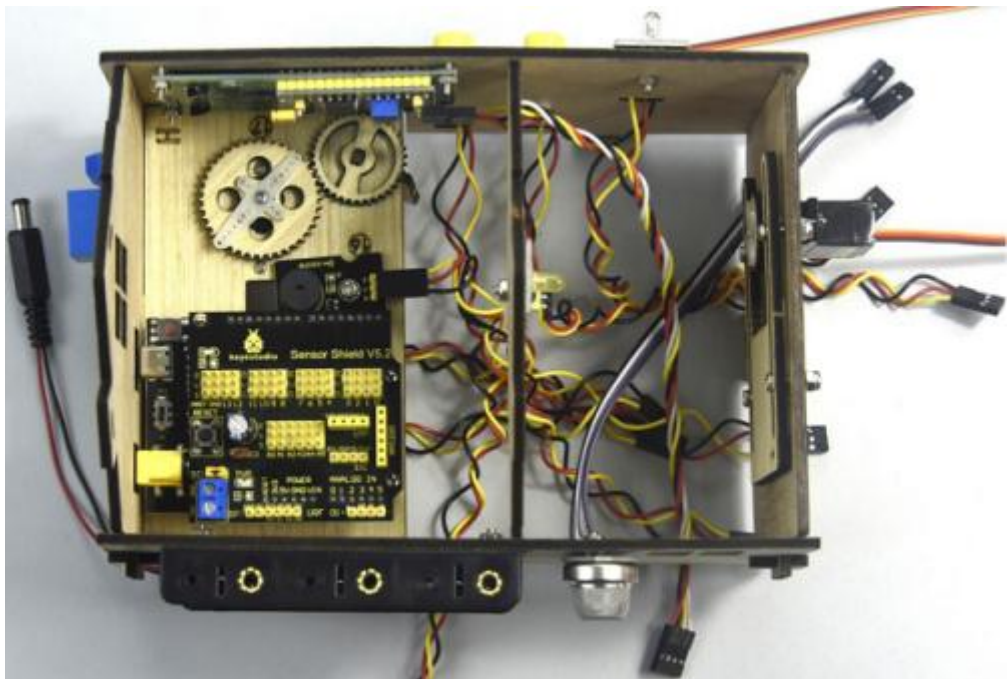
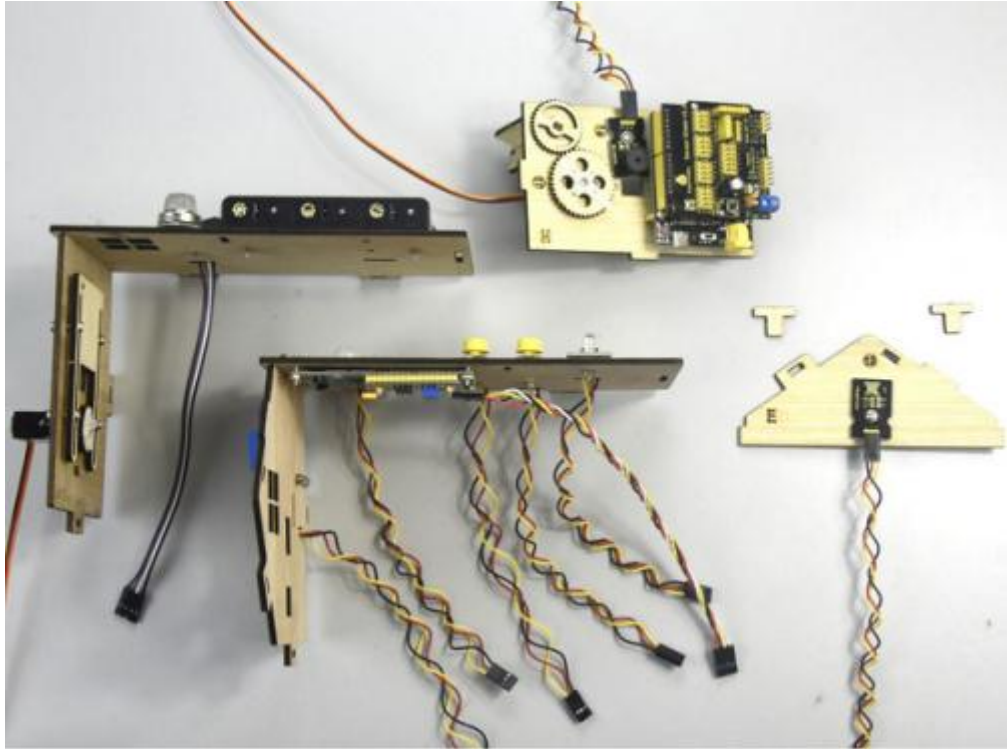


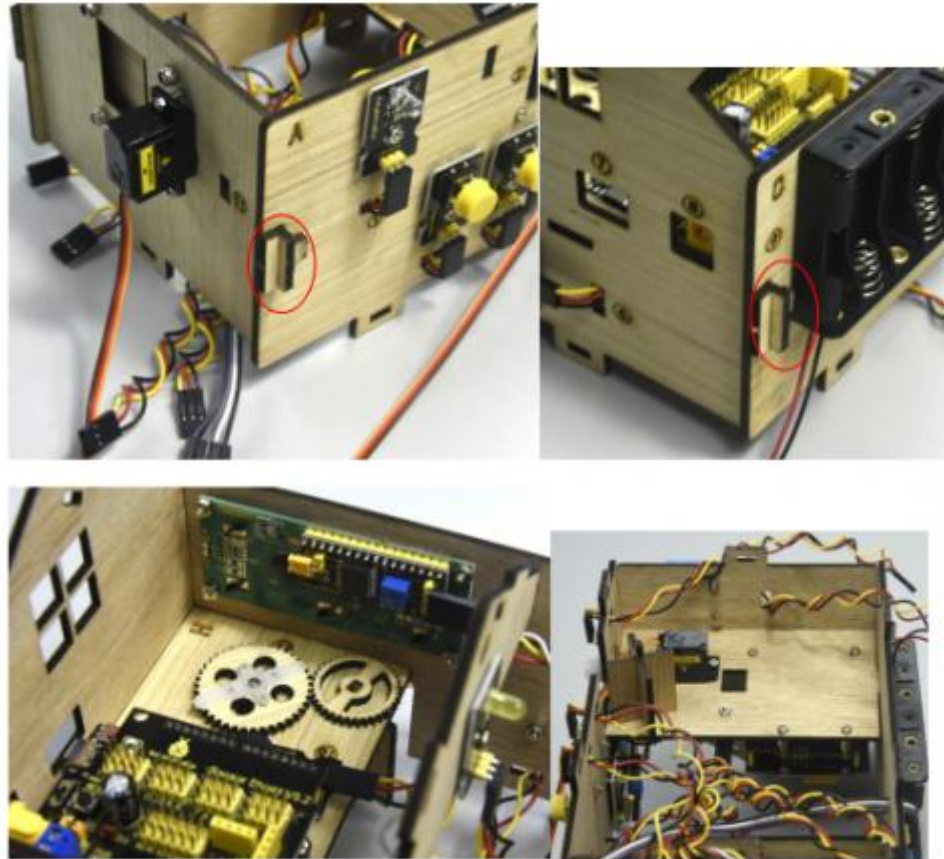
Далі закріплюємо шестерню за допомогою білого хрестового кріплення на чорному сервоприводі за допомогою 1 гвинта M2*5MM (входить до складу сервоприводу), потім встановлюємо комбінацію з 2 плат 2 та іншого сервоприводу на відповідній ділянці плати Н, накладаємо екран датчика на keyestudio плати управління.



Крок 9: Зберання разом плат А, В, С, D, Е і Н, а потім закріплення їх 2 болтами типу «Т».

(Під час бирання разом плат інтерфейс живлення плати керування PLUS має збігатися з отвором ⑧ на платі В, а інтерфейс кабелю USB – з отвором ⑦ на платі В)



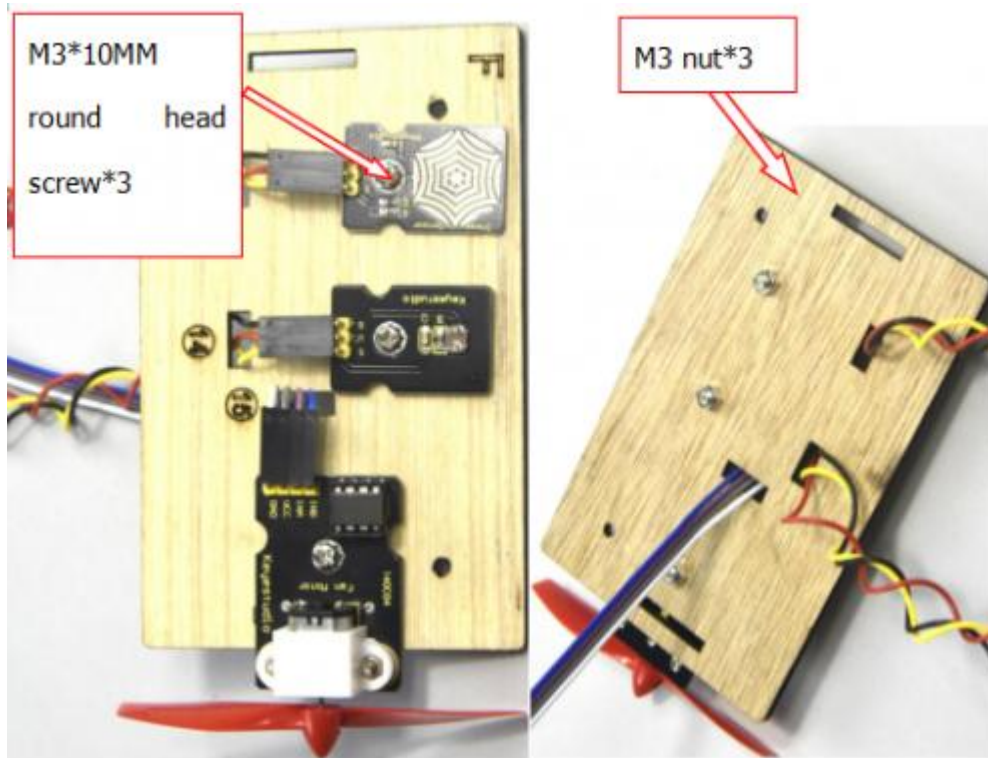


Крок 10: Встановлення датчику плати F

Необхідно підготувати steam sensor, a photocell sensor, a fan module(with fan), a board F, 2pcs 3pinF-F Dupont line, 4pcs F-F dupont lines, 3pcs M3*10MM round head screws and 3pcs M3 nickel plated nuts.

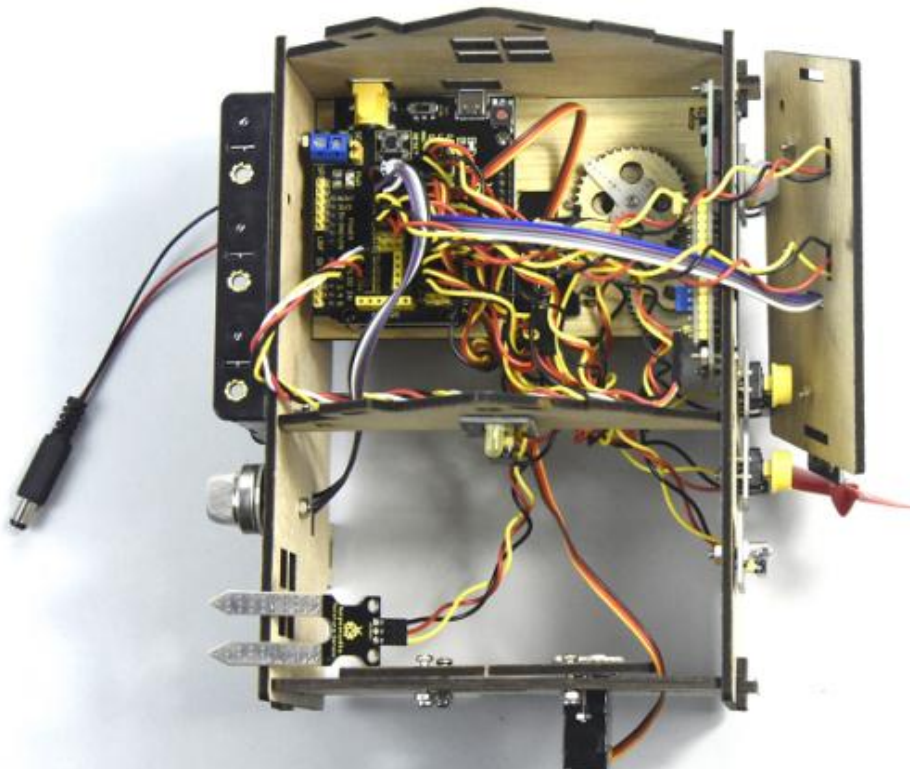
F board*1	Steam sensor*1	Photocell sensor*1	Fan module*1	M3*10MM Round head screw*3	M3 Nickel plated nut*3	F-F Dupont line*4	3pin F-F Dupont line*2

Окремо необхідно закріпити датчик пари, фотоелемент та модуль вентилятора на платі F за допомогою 3 шт. гвинтів M3*10 мм із круглою головкою та 3 шт. гайок M3, а потім з'єднати їх за допомогою 3-контактних та 4-контактних ліній Dupont.

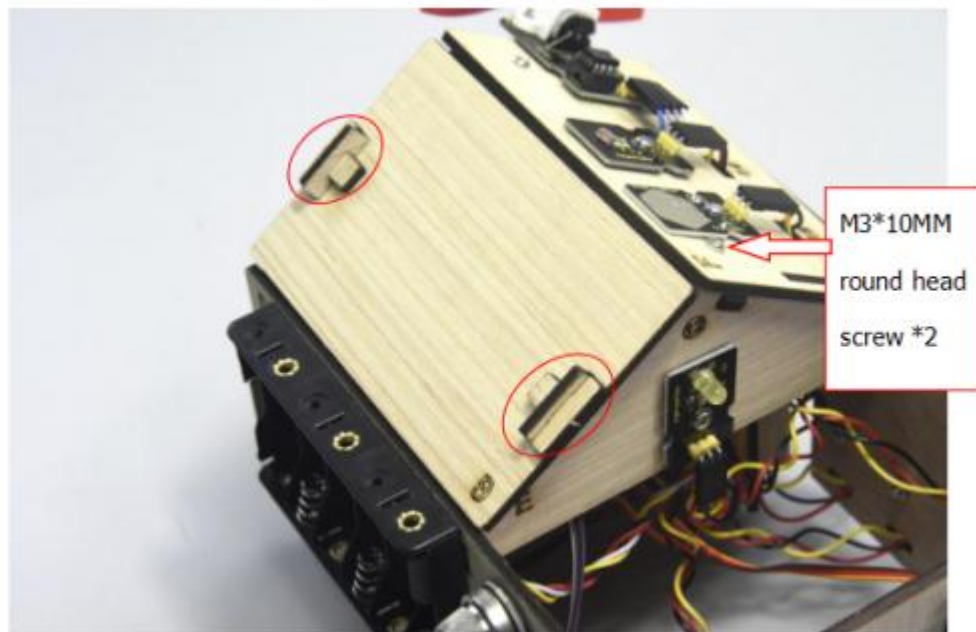


Крок 11: Підключення датчиків/модулів

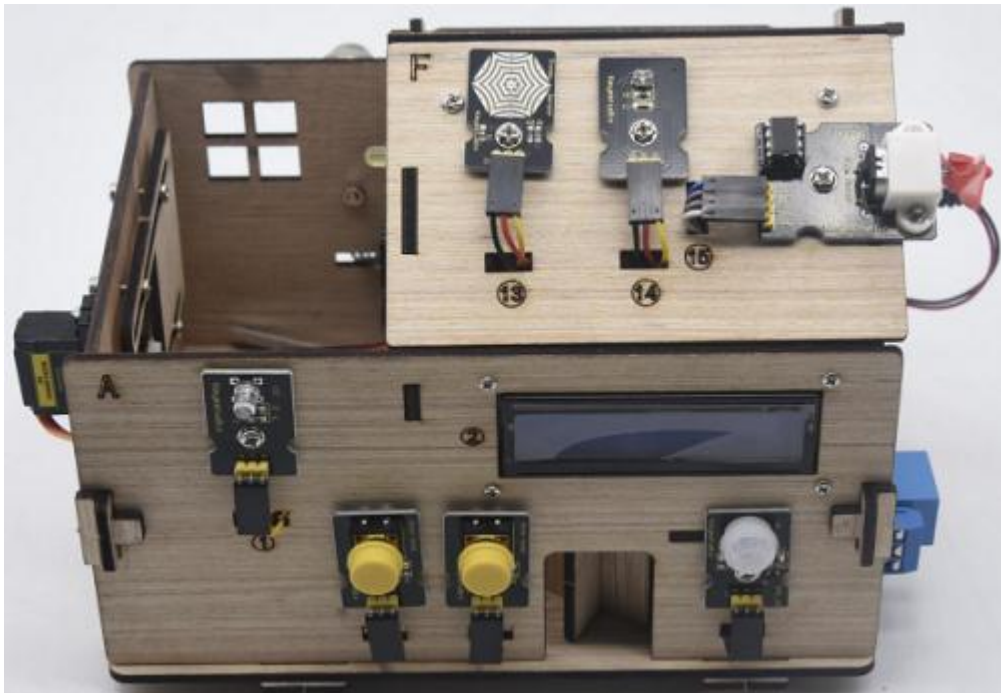
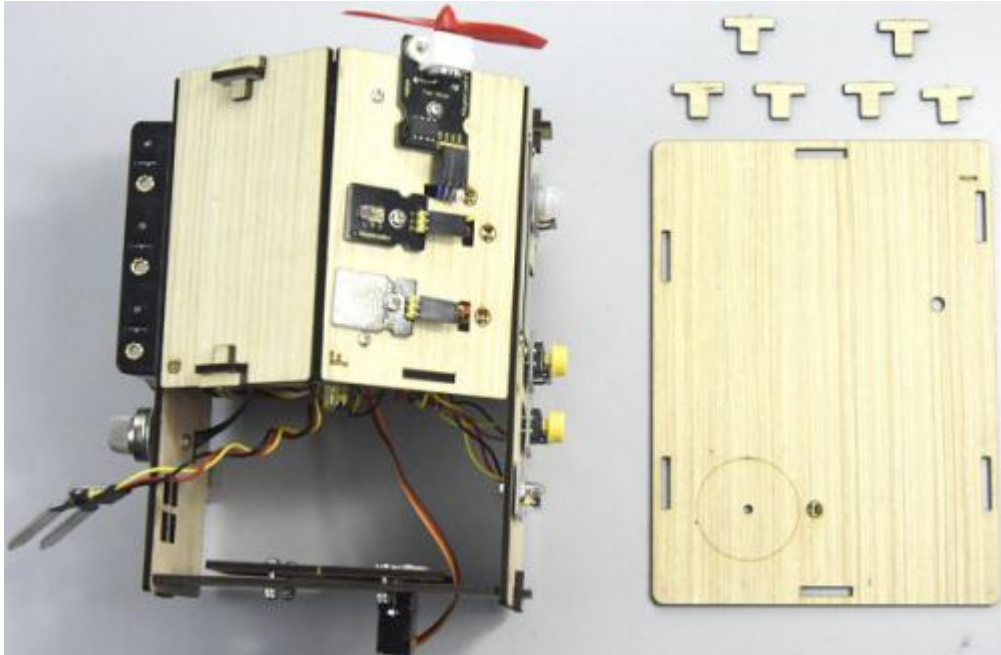
Підключаємо один кінець 3-контактної лінії Dupont до контакту датчика вологості ґрунту, а потім під'єднуємо всі датчики до дисплею.



Вставляємо модуль Bluetooth у екран датчика, потім фіксуємо плату F за допомогою 2 гвинтів M3*10MM з круглою головкою, 2 гайок M3 і 2 частин з отворами посередині, добре закріплюємо плату G за допомогою 2 болтів типу «Т».



Крок 12: Збираємо Smart Home



3. МЕТОДИКА ПРОВЕДЕННЯ ІНТЕГРОВАНИХ УРОКІВ ТЕХНОЛОГІЙ З ВИКОРИСТАННЯМ ARDUINO

3.1 Основні переваги інтегрованих уроків у профільній школі

В даний час актуальна проблема інтеграції, узагальнення предметних знань учнів. Це необхідно для глибшого розуміння процесів і явищ, що відбуваються навколо. Міжпредметна інтеграція - використання теорії, основних понять та визначень, законів та методів одного навчального предмета при вивченні іншого. На даному рівні наявність систематизованого змісту призводить до наступних пізнавальних результатів: відбувається формування цілісної картини світу в уявленні та свідомості учнів та розширення світогляду та кругозору в галузі інтегрованих дисциплін [3]. Як наслідок у школярів з'являються знання нового типу, які знаходять вираження у загальнонаукових поняттях, категоріях, методах та підходах.

Виділимо переваги інтегрованих уроків на прикладі інтеграції технологій та інформатики.

1. Перехід від внутрішньопредметних зв'язків до міжпредметних дозволяють учню перенести способи дій і методи з одних об'єктів на інші, що може полегшити навчання і сформувати у школярів повнішу і цілісну картину світу. Наприклад, щодо теми «Електричні ланцюги» учні зможуть застосувати знання, отримані не лише на уроках технологій, а й інформатики, якщо провести інтеграцію даної теми з курсом конструювання та програмування та дати учням можливість зібрати електричні ланцюги за допомогою Arduino. Arduino є апаратним та програмним засобами для створення простих систем робототехніки та конструювання, які орієнтовані на користувачів аматорського чи непрофесійного рівня [7].

2. Збільшення кількості проблемних ситуацій у структурі інтеграції дисциплін сприятиме активізації мисленнєвої діяльності учнів та підштовхне

їх до пошуку нових способів вивчення навчального матеріалу. Як наслідок відбувається формування дослідницького типу особистості.

3. Завдяки інтеграції відбувається збільшення частки узагальнюючих знань, що дозволяє учню одночасно простежити весь процес виконання дій від мети до результату, осмислено сприймати кожен етап роботи.

Наприклад: на інтегрованому уроці (інформатика та технології) на тему «Електричні двигуни» під час створення презентації для доповіді учні згадують, як працювати у програмі PowerPoint, копіювати, вирізати та вставляти текст та графічні елементи, тим самим закріплюють знання, отримані на уроках інформатики та технологій.[8]

4. Інтеграція підвищує інформативну ємність уроку.

5. Інтеграція дозволяє знаходити нові чинники, які підтверджують чи поглиблюють певні спостереження, висновки учнів щодо різних предметів.

3.2 Методика використання Arduino на інтегрованих уроках у профільній школі

У зв'язку з актуальністю інтеграції технологій та інформатики, можна запропонувати методику інтегрованих уроків з інформатики та технологій з використанням Arduino. У цій методиці приділяється особлива увага практичним заняттям із застосуванням електронних схем на базі Arduino. Електронні схеми являють собою одну або кілька макетних плат, до яких можна підключати елементи без використання спайки або скручування з'єднувальних проводів.[14]

Складання електронної схеми на базі Arduino вимагає наявності у учнів знань найпростіших електронних пристроїв та навичок програмування в різних середовищах, так як для управління платою та підключеними до неї апаратними та механічними пристроями необхідна наявність програмного блоку.

У курсі інформатики учні займаються вивченням тем і понять, які стануть основними в інтегрованому курсі уроків з інформатики та технологій та дозволять підкреслити важливу роль інформатики в даній методиці, тому що для роботи електронних схем Arduino нам потрібне програмне забезпечення. Ось деякі з подібних тем:

- архітектура комп'ютера;
- материнська (системна плата) та процесор;
- види програмного забезпечення;
- операційні системи;
- мови програмування та їх середовища;
- основні етапи вирішення завдань;
- типи даних, операції та функції;
- мова програмування.

У курсі Технологій учні вивчають теми та поняття, які стануть доповнюючими та допоміжними при використанні методики інтегрованого вивчення інформатики та Технологій. Ці теми дозволять пояснити учням принцип роботи електронної схеми та її елементів з погляду Технологій та фізичних явищ:

- електричний струм;
- електричні ланцюги та їх складові частини;
- сила струму та напруга в ланцюгу;
- опір провідників;
- потужність та робота електричного струму;
- електромагнітні явища;
- електромагнітна індукція;
- отримання змінного електричного струму.

Основу методики складає розробка типових електронних пристроїв на базі платформи Arduino. Отриманий досвід може бути корисний учням у

виконанні навчальних проектів. Зміст практичних робіт і види проектів можуть уточнюватися в залежності від схильностей учнів, наявності матеріалів, засобів та ін. Дана методика реалізується у взаємозв'язку з предметами шкільного циклу.[13]

Використання методики орієнтовані на розширений варіант навчання інформатики в основній школі з включенням блоку тем з робототехніки. В результаті застосування даної методики учні мають навчитися:

- розбиратися у схемах та програмних кодах для роботи з мікроелектронними пристроями;
- розробляти та програмувати типові електронні пристрої на базі платформи Arduino;
- керувати складними робототехнічними системами;
- використовувати власні робототехнічні розробки в дослідженнях.

Зміст інтегрованих уроків має включати п'ять розділів, у яких поєднуються поняття з технологій та інформатики для цілісного сприйняття учнями навчального матеріалу.

1. Основні завдання та поняття робототехніки. Предмет робототехніки. Електричні схеми. Електронні пристрої та перетворювачі. Логічні елементи та перемикальні схеми.

2. Моделювання базових електронних пристроїв. Логічні основи цифрових електронних пристроїв. Схемна реалізація логічних функцій. Моделювання тригера на базі Arduino. Моделювання лічильника. Моделювання реверсивного лічильника

3. Робота з цифровими та аналоговими сенсорами. Отримання даних із аналогових сенсорів. Підключення цифрових датчиків до Arduino.

4. Моделювання електронних пристроїв із зворотним зв'язком. Поняття зворотний зв'язок та його роль системах управління.

5. Робота над індивідуальним проектом. Вибір теми проекту, обґрунтування актуальності, постановка мети та завдань, реалізація програмної та технологічної частини проекту.

Як тема індивідуального проекту, наприклад, може бути запропонована «Програмне керування електронними компонентами» та інші. У ході виконання таких проектів учні закріплюють отримані знання, узагальнюють навчальний матеріал з різних шкільних предметів. Інтеграція є джерелом знаходження нових зв'язків між фактами, що підтверджують чи поглиблюють певні висновки спостереження учнів у різних предметах.

ВИСНОВКИ

Виконана кваліфікаційна робота передбачала розробку й виготовлення моделі Smart Home на базі Arduino та методика його використання при вивченні технології у профільній школі. За її результатами можна зробити наступні висновки:

1. Досліджено особливості застосування програмно-апаратної платформи Arduino в навчальному процесі. Визначено можливості програмно-апаратної платформи Arduino в освітньому процесі. Встановлено що платформу Arduino можна використовувати в навчальному процесі для створення проєктів з дослідження мікроконтролерів та технології розробки додатків, що відрізняються високою функціональністю, зручним середовищем програмування та низькою вартістю, зумовленою відкритістю схем та програмного забезпечення.

2. Розроблено ескізний проєкт Smart Home на базі Arduino. Визначено структуру розробленого проєкту, а саме він має включати такі етапи: Формулювання завдання; Вибір способів реалізації алгоритму; Вибір обладнання; Вибір контролера Arduino; Вибір периферії; Розробка принципової схеми; Робота з програмним забезпеченням; Програмування скетчів.

3. Виготовлена модель Smart Home на базі Arduino.

4. Описана методика проведення інтегрованих уроків технологій з використанням Arduino. Визначено основні переваги інтегрованих уроків у профільній школі, представлена методика використання Arduino на інтегрованих уроках у профільній школі. Серед переваг можна виділити наступні: 1. Перехід від внутрішньопредметних зв'язків до міжпредметних дозволяють учню перенести способи дій і методи з одних об'єктів на інші, що може полегшити навчання і сформувати у школярів повнішу і цілісну

картину світу. 2. Збільшення кількості проблемних ситуацій у структурі інтеграції дисциплін сприятиме активізації мисленнєвої діяльності учнів та підштовхне їх до пошуку нових способів вивчення навчального матеріалу. 3. Завдяки інтеграції відбувається збільшення частки узагальнюючих знань, що дозволяє учню одночасно простежити весь процес виконання дій від мети до результату, осмислено сприймати кожен етап роботи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бондар В. І. Дидактика: навч. посіб. Київ: Либідь, 2005. 264 с.
2. Биков В. Ю. Інформаційні технології і засоби навчання. Київ: Атіка, 2008. 684 с.
3. Вакуленко І. В. Управління самостійною роботою студентів з використанням інформаційно-комунікаційних технологій. *Науковий часопис Національного педагогічного університету імені М. П. Драгоманова. Серія 2: Комп'ютерно-орієнтовані системи навчання*. Київ. 2016. Вип. 18 (25). С. 50–64.
4. Гервас О. Г. Ергономіка: навч.-метод. посібник. Умань: Видавничо-поліграфічний центр «Візаві», 2011. 130 с.
5. Горбатюк Р. М. Формування проектної діяльності майбутніх інженерів-педагогів комп'ютерного профілю. *Педагогічні науки : зб. наукових праць Херсонського держ. ун-ту*. Херсон, 2009. Вип. 52. С. 433–439.
6. Кривонос О. М. Огляд платформи Arduino Nano 3.0 та перспективи використання під час навчального процесу. Інформаційні технології і засоби навчання. Том 56. № 6. Київ. 2016. С.79–80.
7. Кабак В. В. Модель підготовки майбутніх інженерів-педагогів технічного університету до професійної діяльності засобами комп'ютерних технологій. *Нова педагогічна думка: Науково-методичний журнал*. 2013. № 3 (75). С. 63–66.
8. Лещук С. О. Навчально-інформаційне середовище як засіб організації пізнавальної діяльності учнів. *Науковий часопис Національного педагогічного університету ім. М. П. Драгоманова. Серія 2: Комп'ютерно-орієнтовані системи навчання: До 170-річного ювілею*. Київ: НПУ, 2004. С. 305–313.

9. Мусійовська О. Ф. Моделі комбінованого навчання. *Інформаційно-телекомунікаційні технології в сучасній освіті: досвід, проблеми, перспективи*: Львів: ЛДУ БЖД, 2009. Ч. 2. С. 313–318.
10. Онлайн курс Arduino на базі простого стартового набору [Електронний ресурс]. Geektimes. Режим доступу : <https://geektimes.ru/post/279860/> (дата звернення: 20.11.2023).
11. Ортинський В. Л. Педагогіка вищої школи: навч. посіб. для студ. вищ. навч. закл. Київ: Центр учбової літератури, 2009. 472 с.
12. Остапчук С.А. До проблеми використання платформи Arduino у вивченні робототехніки. Наукові записки ЦДПУ. Серія: Педагогічні науки. Кропивницький: РВВ ЦДПУ ім. В. Винниченка, 2018. Вип. 168. С. 178-181.
13. Петрицин І. О. Застосування комп'ютерного моделювання у процесі електротехнічної підготовки майбутнього вчителя технологій. *Молодь і ринок*. 2017. № 1. С. 60–64.
14. Плата Arduino Uno R3: схема, опис, підключення пристроїв [Електронний ресурс]. ArduinoMaster. Режим доступу: arduinomaster.ru/platy-arduino/plata-arduino-uno/#__Arduino_Uno_R3(дата звернення: 20.11.2023).
15. Програмування Arduino [Електронний ресурс]. Arduino. Режим доступу: arduino.ru/Reference (дата звернення: 20.11.2023).
16. Райковська Г.О., Соловійов А.В. Особливості використання САЕ-систем у навчальному процесі майбутніх бакалаврів з механічної інженерії. Науковий вісник Ужгородського університету. Серія: «Педагогіка. Соціальна робота». 2017. Випуск 2 (41). С.216–218.
17. Триус Ю. В., Герасименко І. В. Комбіноване навчання як інноваційна освітня технологія у вищій школі. *Theory and methods of e-learning*. 2012. Т. 3. С. 299–308.

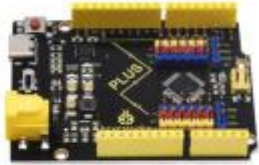



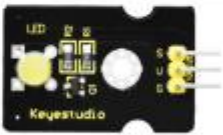

18. Умрик М. А. Організація самостійної роботи майбутніх учителів засобами інформаційно-навчального середовища. *Проблеми сучасної педагогічної освіти. Сер. : Педагогіка і психологія: зб. статей.* Ялта: РВВ КГУ, 2007. Вип.16. Ч. 1. С. 88–95.
19. Andersen, B.B. (1999). The Art of Seeing the Wood and the Trees: Teachers' New Competencies in Terms of Multimedia Literacy and ICT Genre Didactical Competencies. Copenhagen, Royal Danish School of Educational Studies Research Centre for Education and ICT.
20. Arduino Products [Електронний ресурс]. Режим доступу: <https://www.arduino.cc/en/Main/Products> (дата звернення: 20.11.2023).
21. Kushner D. The Making of Arduino [Електронний ресурс]. Режим доступу: <https://web.eecs.umich.edu/~prabal/teaching/resources/eecs582/kushner11arduino.pdf> (дата звернення: 20.11.2023).
22. Margolis Michael. Arduino Cookbook. 2nd Edition. O'Reilly, 2011. 724 p.
23. Microcontroller [Електронний ресурс]. Режим доступу: <https://www.techopedia.com/definition/3641/microcontroller> (дата звернення: 20.11.2023).
24. Nano Плати Ардуіно [Електронний ресурс]. Режим доступу: <http://arduino.ua/ru/hardware/Nano> (дата звернення: 20.11.2023).









ДОДАТКИ










Додаток А










Проект: Багатоцільовий розумний будинок





Перелік компонентів

#	Name	QTY	Picture
1	PLUS Control Board (Compatible with <u>Arduino</u> UNO)	1	
2	<u>keyestudio</u> Sensor Shield	1	
3	Wooden Board*10T=3MM	1	
4	White LED Module	1	
5	Yellow LED Module	1	
6	Button Sensor	2	

7	Photocell Sensor	1	 A black PCB module with a circular photocell sensor in the center, a potentiometer, and three gold-plated pins on the right. The brand name 'Keyestudio' is printed at the bottom.
8	PIR Motion Sensor	1	 A black PCB module with a white hemispherical PIR sensor lens, a potentiometer, and three gold-plated pins on the right. The brand name 'Keyestudio' is printed at the bottom.
9	MQ-2 Gas Sensor	1	 A black PCB module with a circular MQ-2 gas sensor, a potentiometer, and three gold-plated pins on the right. The brand name 'Keyestudio' is printed at the bottom.
10	Relay Module	1	 A blue PCB module with a relay component, a potentiometer, and three gold-plated pins on the right. The brand name 'keyestudio' is printed at the top.
11	Bluetooth HM-10 Module	1	 A blue PCB module with a Bluetooth module, a potentiometer, and three gold-plated pins on the right.
12	Passive Buzzer Sensor	1	 A black PCB module with a circular passive buzzer, a potentiometer, and three gold-plated pins on the right. The brand name 'Keyestudio' is printed at the bottom.
13	Fan module	1	 A small black PCB module with a fan and a potentiometer. A red fan blade is visible on the right side.
14	Steam Sensor	1	 A black PCB module with a steam sensor, a potentiometer, and three gold-plated pins on the right. The brand name 'Keyestudio' is printed at the bottom.

15	Servo Motor	2	
16	LCD1602 Display Module	1	
17	Soil Humidity Sensor	1	
18	USB Cable	1	
19	Female to Female Dupont lines	40	
20	Male to female dupont line	6	
21	M3 Nickel Plated Nut	25	
22	M2*12MM Round Head Screw	6	
23	M2 Nickel Plated Nut	6	

24	M3*10MM Dual-pass Copper Pillar	4	
25	M3*6MM Round Head Screw	8	
26	M3 304 Stainless Steel Self-locking Nut	4	
27	M3*10MM Round Head Screw	20	
28	M2.5*10MM Round Head Screw	6	
29	M2.5 Nickel Plated nut	6	
30	M3*12MM Round Head Screw	6	
31	M3*10MM Flat Head Screw	2	
32	M1.2*5MM Round Head Self-tapping Screw	10	

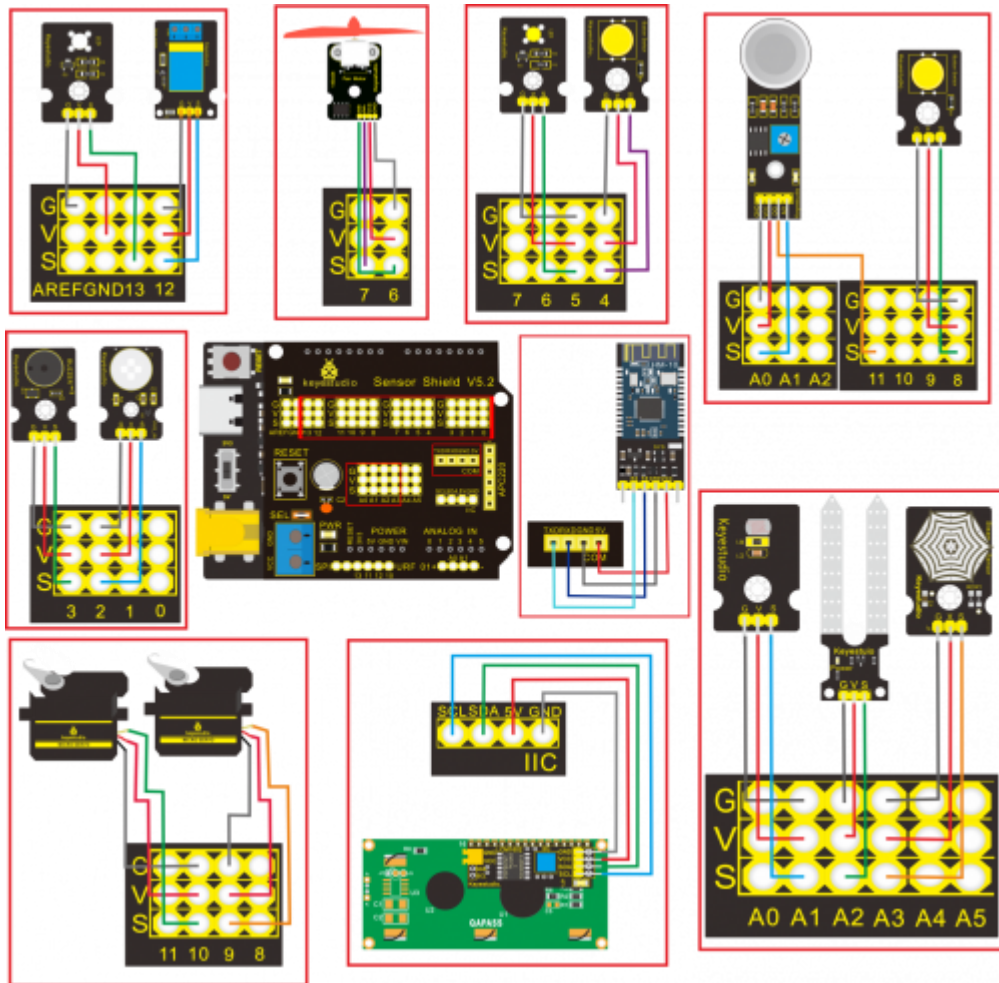
33	6-cell AA battery holder with DC head and 15cm dew line	1	
34	Black-yellow Handle 3*40MM Cross Screwdriver	1	
35	2.54 3Pin F-F jumper wire 20cm	13	
36	2.54 4Pin F-F jumper wire 20cm	2	

Проект: Багатоцільовий розумний будинок

Обладнання:

Keyestudio PLUS control board	Sensor shield	Fan module*1	Servo motor*2	LCD1602 display*1	Button sensor*2	White LED*1
Relay module*1	Passive buzzer*1	PIR motion sensor*1	Steam sensor*1	photo cell sensor*1	Bluetooth module*1	Yellow LED*1
Soil humidity sensor*1	MQ-2 Gas sensor*1	4pin F-F Dupont line*1	F-F Dupont lines	USB cable*1	3pin F-F Dupont line*10	

Схема підключення:



Проект: Багатоцільовий розумний будинок

Тестовий код:

```
//call the relevant library file
#include <Servo.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
//Set the communication address of I2C to 0x27, display 16 characters
every line, two lines in total
LiquidCrystal_I2C mylcd(0x27, 16, 2);

//set ports of two servos to digital 9 and 10
Servo servo_10;
Servo servo_9;

volatile int btn1_num;//set variable btn1_num
volatile int btn2_num;//set variable btn2_num
volatile int button1;//set variable button1
volatile int button2;//set variable button2
String fans_char;//string type variable fans_char
volatile int fans_val;//set variable fans_char
volatile int flag;//set variable flag
volatile int flag2;//set variable flag2
volatile int flag3;//set variable flag3
volatile int gas;//set variable gas
volatile int infrar;//set variable infrar
String led2;//string type variable led2
volatile int light;//set variable light
String pass;//string type variable pass
String passwd;//string type variable passwd

String servo1;//string type variable servo1
volatile int servo1_angle;//set variable light
String servo2;//string type variable servo2
volatile int servo2_angle;//set variable servo2_angle

volatile int soil;//set variable soil
volatile int val;//set variable val
volatile int value_led2;//set variable value_led2
volatile int water;//set variable water

int length;
int tonepin = 3; //set the signal end of passive buzzer to digital 3
//define name of every sound frequency
#define D0 -1
#define D1 262
#define D2 293
#define D3 329
#define D4 349
#define D5 392
#define D6 440
#define D7 494
#define M1 523
```

```

#define M2 586
#define M3 658
#define M4 697
#define M5 783
#define M6 879
#define M7 987
#define H1 1045
#define H2 1171
#define H3 1316
#define H4 1393
#define H5 1563
#define H6 1755
#define H7 1971

#define WHOLE 1
#define HALF 0.5
#define QUARTER 0.25
#define EIGHTH 0.25
#define SIXTEENTH 0.625

//set sound play frequency
int tune[] =
{
    M3, M3, M4, M5,
    M5, M4, M3, M2,
    M1, M1, M2, M3,
    M3, M2, M2,
    M3, M3, M4, M5,
    M5, M4, M3, M2,
    M1, M1, M2, M3,
    M2, M1, M1,
    M2, M2, M3, M1,
    M2, M3, M4, M3, M1,
    M2, M3, M4, M3, M2,
    M1, M2, D5, D0,
    M3, M3, M4, M5,
    M5, M4, M3, M4, M2,
    M1, M1, M2, M3,
    M2, M1, M1
};

//set music beat
float durst[] =
{
    1, 1, 1, 1,
    1, 1, 1, 1,
    1, 1, 1, 1,
    1 + 0.5, 0.5, 1 + 1,
    1, 1, 1, 1,
    1, 1, 1, 1,
    1, 1, 1, 1,
    1 + 0.5, 0.5, 1 + 1,
    1, 1, 1, 1,
    1, 0.5, 0.5, 1, 1,
    1, 0.5, 0.5, 1, 1,
    1, 1, 1, 1,

```

```

    1, 1, 1, 1,
    1, 1, 1, 0.5, 0.5,
    1, 1, 1, 1,
    1 + 0.5, 0.5, 1 + 1,
};

void setup() {
  Serial.begin(9600); //set baud rate to 9600

  mylcd.init();
  mylcd.backlight(); //initialize LCD
  //LCD shows "passcord:" at first row and column
  mylcd.setCursor(1 - 1, 1 - 1);
  mylcd.print("passcord:");

  servo_9.attach(9); //make servo connect to digital 9
  servo_10.attach(10); //make servo connect to digital 10
  servo_9.write(0); //set servo connected digital 9 to 0°
  servo_10.write(0); //set servo connected digital 10 to 0°
  delay(300);

  pinMode(7, OUTPUT); //set digital 7 to output
  pinMode(6, OUTPUT); //set digital 6 to output
  digitalWrite(7, HIGH); //set digital 7 to high level
  digitalWrite(6, HIGH); //set digital 6 to high level

  pinMode(4, INPUT); //set digital 4 to input
  pinMode(8, INPUT); //set digital 8 to input
  pinMode(2, INPUT); //set digital 2 to input
  pinMode(3, OUTPUT); //set digital 3 to output
  pinMode(A0, INPUT); //set A0 to input
  pinMode(A1, INPUT); //set A1 to input
  pinMode(13, OUTPUT); //set digital 13 to input
  pinMode(A3, INPUT); //set A3 to input
  pinMode(A2, INPUT); //set A2 to input

  pinMode(12, OUTPUT); //set digital 12 to output
  pinMode(5, OUTPUT); //set digital 5 to output
  pinMode(3, OUTPUT); //set digital 3 to output
  length = sizeof(tune) / sizeof(tune[0]); //set the value of length
}

void loop() {
  auto_sensor();
  if (Serial.available() > 0) //serial reads the characters
  {
    val = Serial.read(); //set val to character read by serial
    Serial.println(val); //output val character in new lines
    pwm_control();
  }
  switch (val) {
    case 'a': //if val is character 'a', program will circulate
      digitalWrite(13, HIGH); //set digital 13 to high level, LED
      lights up
      break; //exit loop
  }
}

```



```

case 'b'://if val is character 'b', program will circulate
    digitalWrite(13, LOW); //Set digital 13 to low level, LED is off
    break;//exit loop
case 'c'://if val is character 'c', program will circulate
    digitalWrite(12, HIGH); //set digital 12 to high level, NO of relay
is connected to COM
    break;//exit loop
case 'd'://if val is character 'd', program will circulate
    digitalWrite(12, LOW); //set digital 12 to low level, NO of relay is
disconnected to COM

    break;//exit loop
case 'e'://if val is character 'e', program will circulate
    music1();//play birthday song
    break;//exit loop
case 'f'://if val is character 'f', program will circulate
    music2();//play ode to joy song
    break;//exit loop
case 'g'://if val is character 'g', program will circulate
    noTone(3);//set digital 3 to stop playing music
    break;//exit loop
case 'h'://if val is character 'h', program will circulate
    Serial.println(light);//output the value of variable light in new
lines
    delay(100);
    break;//exit loop
case 'i'://if val is character 'i', program will circulate
    Serial.println(gas);//output the value of variable gas in new lines
    delay(100);
    break;//exit loop
case 'j'://if val is character 'j', program will circulate
    Serial.println(soil);//output the value of variable soil in new
lines
    delay(100);
    break;//exit loop
case 'k'://if val is character 'k', program will circulate
    Serial.println(water);//output the value of variable water in new
lines
    delay(100);
    break;//exit loop
case 'l'://if val is character 'l', program will circulate
    servo_9.write(180);//set servo connected to digital 9 to 180°
    delay(500);
    break;//exit loop
case 'm'://if val is character 'm', program will circulate
    servo_9.write(0);//set servo connected to digital 9 to 0°
    delay(500);
    break;//exit loop
case 'n'://if val is character 'n', program will circulate
    servo_10.write(180);//set servo connected to digital 10 to 180°
    delay(500);
    break;//exit loop
case 'o'://if val is character 'o', program will circulate
    servo_10.write(0);//set servo connected to digital 10 to 0°

```

```

    delay(500);
    break;//exit loop
case 'p'://if val is character 'p', program will circulate
    digitalWrite(5, HIGH); //set digital 5 to high level, LED is on
    break;//exit loop
case 'q'://if val is character 'q', program will circulate
    digitalWrite(5, LOW); // set digital 5 to low level, LED is off
    break;//exit loop
case 'r'://if val is character 'r', program will circulate
    digitalWrite(7, LOW);
    digitalWrite(6, HIGH); //fan rotates anticlockwise at the fastest
speed
    break;//exit loop
case 's'://if val is character 's', program will circulate
    digitalWrite(7, LOW);
    digitalWrite(6, LOW); //fan stops rotating
    break;//exit loop
}
}

////////////////////////////////////set birthday
song////////////////////////////////////
void birthday()
{
    tone(3, 294); //digital 3 outputs 294HZ sound
    delay(250);//delay in 250ms
    tone(3, 440);
    delay(250);
    tone(3, 392);
    delay(250);
    tone(3, 532);
    delay(250);
    tone(3, 494);
    delay(500);
    tone(3, 392);
    delay(250);
    tone(3, 440);
    delay(250);
    tone(3, 392);
    delay(250);
    tone(3, 587);
    delay(250);
    tone(3, 532);
    delay(500);
    tone(3, 392);
    delay(250);
    tone(3, 784);
    delay(250);
    tone(3, 659);
    delay(250);
    tone(3, 532);
    delay(250);
    tone(3, 494);
    delay(250);
    tone(3, 440);
    delay(250);
}

```

```

tone(3, 698);
delay(375);
tone(3, 659);
delay(250);
tone(3, 532);
delay(250);
tone(3, 587);
delay(250);
tone(3, 532);
delay(500);
}

//detect gas
void auto_sensor() {
  gas = analogRead(A0); //assign the analog value of A0 to gas
  if (gas > 700) {
    //if variable gas>700
    flag = 1; //set variable flag to 1
    while (flag == 1)
      //if flag is 1, program will circulate
      {
        Serial.println("danger"); //output "danger" in new lines
        tone(3, 440);
        delay(125);
        delay(100);
        noTone(3);
        delay(100);
        tone(3, 440);
        delay(125);
        delay(100);
        noTone(3);
        delay(300);
        gas = analogRead(A0); //gas analog the value of A0 to gas
        if (gas < 100) //if variable gas is less than 100
          {
            flag = 0; //set variable flag to 0
            break; //exit loop exist to loop
          }
      }
  }
  } else
  //otherwise
  {
    noTone(3); // digital 3 stops playing music
  }
  light = analogRead(A1); //Assign the analog value of A1 to light
  if (light < 300) //if variable light is less than 300
  {
    infrared = digitalRead(2); //assign the value of digital 2 to infrared
    Serial.println(infrared); //output the value of variable infrared in new
lines
    if (infrared == 1)
      // if variable infra is 1
      {

```

```

    digitalWrite(13, HIGH); //set digital 13 to high level, LED is on
  } else//Otherwise
  {
    digitalWrite(13, LOW); //set digital 13 to low level, LED is off
  }
}
water = analogRead(A3);//assign the analog value of A3 to variable water
if (water > 800)
  // if variable water is larger than 800
  {
    flag2 = 1;//if variable flag 2 to 1
    while (flag2 == 1)
      // if flag2 is 1, program will circulate
      {
        Serial.println("rain");//output "rain" in new lines
        servo_10.write(180);// set the servo connected to digital 10 to 180°
        delay(300);//delay in 300ms
        delay(100);
        water = analogRead(A3);;//assign the analog value of A3 to variable
water
        if (water < 30)// if variable water is less than 30
          {
            flag2 = 0;// set flag2 to 0
            break;//exit loop
          }
        }
  } else//Otherwise
  {
    if (val != 'u' && val != 'n')
      //if val is not equivalent 'u' either 'n'
      {
        servo_10.write(0);//set servo connected to digital 10 to 0°
        delay(10);
      }
  }
}
soil = analogRead(A2);//assign the analog value of A2 to variable soil
if (soil > 50)
  // if variable soil is greater than 50
  {
    flag3 = 1;//set flag3 to 1
    while (flag3 == 1)
      //If set flag3 to 1, program will circulate
      {
        Serial.println("hydropenia ");//output "hydropenia " in new lines
        tone(3, 440);
        delay(125);
        delay(100);
        noTone(3);
        delay(100);
        tone(3, 440);
        delay(125);
        delay(100);
      }
  }
}

```

```

    noTone(3); //digital 3 stops playing sound
    delay(300);
    soil = analogRead(A2); //Assign the analog value of A2 to variable
soil
    if (soil < 10) //If variable soil < 10
    {
        flag3 = 0; //set flag3 to 0
        break; //exit loop
    }
} else //Otherwise
{
    noTone(3); //set digital 3 to stop playing music
}
door(); //run subroutine
}

void door() {
    button1 = digitalRead(4); // assign the value of digital 4 to button1
    button2 = digitalRead(8); //assign the value of digital 8 to button2

    if (button1 == 0) //if variable button1 is 0
    {
        delay(10); //delay in 10ms
        while (button1 == 0) //if variable button1 is 0, program will circulate
        {
            button1 = digitalRead(4); // assign the value of digital 4 to button1
            btn1_num = btn1_num + 1; //variable btn1_num plus 1
            delay(100); // delay in 100ms
        }
    }
    if (btn1_num >= 1 && btn1_num < 5) //1 ≤ if variable btn1_num < 5
    {
        Serial.print(".");
        Serial.print("");
        passwd = String(passwd) + String("."); //set passwd
        pass = String(pass) + String("."); //set pass
        //LCD shows pass at the first row and column
        mylcd.setCursor(1 - 1, 2 - 1);
        mylcd.print(pass);
    }
    if (btn1_num >= 5)
        //if variable btn1_num ≥ 5
    {
        Serial.print("-");
        passwd = String(passwd) + String("-"); //Set passwd
        pass = String(pass) + String("-"); //set pass
        //LCD shows pass at the first row and column
        mylcd.setCursor(1 - 1, 2 - 1);
        mylcd.print(pass);
    }
}
if (button2 == 0) //if variable button2 is 0

```

```

{
  delay(10);
  if (button2 == 0)//if variablebutton2 is 0
  {
    if (passwd == ".--.-.")//if passwd is ".--.-."
    {
      mylcd.clear();//clear LCD screen
      //LCD shows "open!" at first character on second row
      mylcd.setCursor(1 - 1, 2 - 1);
      mylcd.print("open!");
      servo_9.write(100);//set servo connected to digital 9 to 100°
      delay(300);
      delay(5000);
      passwd = "";
      pass = "";
      mylcd.clear();//clear LCD screen
      //LCD shows "password:" at first character on first row
      mylcd.setCursor(1 - 1, 1 - 1);
      mylcd.print("password:");

    } else //Otherwise
    {
      mylcd.clear();//clear LCD screen
      //LCD shows "error!" at first character on first row
      mylcd.setCursor(1 - 1, 1 - 1);
      mylcd.print("error!");
      passwd = "";
      pass = "";
      delay(2000);
      //LCD shows "again" at first character on first row
      mylcd.setCursor(1 - 1, 1 - 1);
      mylcd.print("again");
    }
  }
}
infrar = digitalRead(2);//assign the value of digital 2 to infrar
if (infrar == 0 && (val != 'l' && val != 't'))
  //if variable infrar is 0 and val is not 'l' either 't'
  {
    servo_9.write(0);//set servo connected to digital 9 to 0°
    delay(50);
  }
if (button2 == 0)//if variablebutton2 is 0
{
  delay(10);
  while (button2 == 0) //if variablebutton2 is 0, program will circulate
  {
    button2 = digitalRead(8);//assign the value of digital 8 to button2
    btn2_num = btn2_num + 1;//variable btn2_num plus 1
    delay(100);
    if (btn2_num >= 15)//if variablebtn2_num ≥15
    {
      tone(3, 532);
      delay(125);
      mylcd.clear();//clear LCD screen
      //LCD shows "password:" at the first character on first row

```

```

        mylcd.setCursor(1 - 1, 1 - 1);
        mylcd.print("password:");
        //LCD shows "wait" at the first character on first row
        mylcd.setCursor(1 - 1, 1 - 1);
        mylcd.print("wait");
    } else//Otherwise
    {
        noTone(3);//digital 3 stops playing music
    }
}

}

btn1_num = 0;//set btn1_num to 0
btn2_num = 0;//set btn2_num to 0
}

// Birthday song
void music1() {
    birthday();
}
//Ode to joy
void music2() {
    Ode_to_Joy();
}
void Ode_to_Joy()//play Ode to joy song
{
    for (int x = 0; x < length; x++)
    {
        tone(tonepin, tune[x]);
        delay(300 * durt[x]);
    }
}

//PWM control
void pwm_control() {
    switch (val)
    {
        case 't'://if val is 't', program will circulate
            servo1 = Serial.readStringUntil('#');
            servo1_angle = String(servo1).toInt();
            servo_9.write(servo1_angle);//set the angle of servo connected to
digital 9 to servo1_angle
            delay(300);
            break;//exit loop
        case 'u'://if val is 'u', program will circulate
            servo2 = Serial.readStringUntil('#');
            servo2_angle = String(servo2).toInt();
            servo_10.write(servo2_angle);//set the angle of servo connected to
digital 10 to servo2_angle
            delay(300);
            break;//exit loop
        case 'v'://if val is 'v', program will circulate
            led2 = Serial.readStringUntil('#');
            value_led2 = String(led2).toInt();
            analogWrite(5, value_led2); //PWM value of digital 5 is value_led2
            break;//exit loop
    }
}

```

```
case 'w': //if val is 'w', program will circulate
  fans_char = Serial.readStringUntil('#');
  fans_val = String(fans_char).toInt();
  digitalWrite(7, LOW);
  analogWrite(6, fans_val); //set PWM value of digital 6 to
fans_val, the larger the value, the faster the fan
  break; //exit loop
}
}
```