Министерство образования Оренбургской области

Муниципальное бюджетное общеобразовательное учреждение

«Средняя общеобразовательная школа № 117 имени М.В. Стрельникова»

города Сорочинска Оренбургской области

**«Модель «Умный дом»»**

Выполнил(а):

Догонкин Степан Николаевич,

обучающийся 10 класса,

МБОУ «СОШ № 117»

г. Сорочинска Оренбургской области,

Руководитель:

Лупова Наталья Александровна,

учитель информатики ,

1 квалификационной категории

МБОУ «СОШ № 117»

г. Сорочинска Оренбургской области

Сорочинск, 2023

**Аннотация**

Проект по робототехнике "Модель «Умный дом»" содержит

справочные сведения о системе "умный дом", а также автор представляет свою разработку модуля системы «умный дом», разработку и исследование алгоритмов системы, позволяющих увеличить экономию ресурсов.

Подробнее о проекте:  
Творческая работа учащегося 10 класса о создании программы управления "умным домом" содержит обзор систем управления зданиями и готовых программных решений, а также описывает выбор аппаратных средств, определяет нюансы, на которые стоит обратить внимание при внедрении программы "умный дом" и ее эксплуатации.

Коды для написания программы управления системой "умный дом" представлены в приложении к проекту.

Оглавление

Введение……………………………………………………………….. 4  
1. Теоретическая часть ……………………………………………….. 6

* 1. **Программа Arduino……………………………………………6**

1. Разработка устройства………………………………………............7
   1. **Этапы создания робота «Умный дом»……………………....7**
   2. **Разработка устройства………………………………………...8**

Выводы и практические рекомендации……………………………..14

Заключение……………………………………………………………15  
Список литературы…………………………………………………...16  
Приложение 1 …………………………………………………………17

Приложение 2 …………………………………………………………18

**Введение**

Само понятие "умный" дом (англ. smart house) не такое уж молодое. Оно возникло в США в начале 70-х годов прошлого века, в недрах "Института интеллектуальных зданий". На тот момент под умным домом подразумевалось "здание, обеспечивающее продуктивное и эффективное использование рабочего пространства…".

Однако годом рождения современного "умного" дома можно считать 1978-й год. В этом году в США компании Х10 USA и Leviton разработали и внедрили в производство технологию управления бытовыми приборами по проводам бытовой электросети.

Распространение эти разработки получили в то время лишь на территории Северной Америки, ибо были рассчитаны на работу при напряжении 110 В и частоту сети 60 Гц. Тем не менее, именно этим фирмам человечество обязано появлением "невероятных чудес прогресса" - автоматически открывающихся дверей, включающегося по хлопку света и прочих "фокусов", которыми состоятельные американцы удивляли своих гостей, а голливудские фильмы - весь остальной мир.

Для конца 70-х годов технология X10 (именно такое название закрепилось и сохранилось за ней и поныне) являлась, конечно, революционной. Однако она была рассчитана на поддержку всего шести управленческих команд и, в основном, использовалась для управления электроосвещением. Но людям хотелось большего. "Умный" дом должен был становиться все "умнее".

С начала нового тысячелетия человечество шагает в эпоху новых технологических открытий, одним из которых является бытовая автоматизация. Время современного человека имеет огромную ценность и такие системы автоматизации как «умный дом» существенно экономят этот жизненно-важный ресурс.

**Актуальность проекта.** Программа Аrduino и mblock5 составлена с учетом тенденций развития современных информационных технологий, что позволяет сохранять актуальность ее реализации.

Все они организованы общим интеллектуальным управлением и им же контролируются. К примеру, человеку не нужно думать об оставленном включенным утюге или электроплите — когда он уйдет, все будет отключено автоматом. Даже не стоит думать, что Intellhouse — это отдельный дом. Интеллектуальной можно сделать и квартиру, главное — разработать соответствующий проект и подобрать оборудование.

Создание удобства и уюта посредством контроля технических процессов — главное предназначение «Умного дома».

**Цель проекта**: изучить процесс управления многофункциональным устройством, на примере умного дома.

**Задачи проекта:**

1. Изучить основы программирования комплектов платы Arduino в комплекте КЛИК.
2. Научиться создавать проекты из комплектов Arduino и mblokc 5.
3. Собрать конструкцию робота, выполняющего функции умного дома.
4. Составить программу для робота, выполняющего функции умного дома.
5. Определить дополнительные возможности изменения конструкции для робота, выполняющего функции умного дома.

**Объект исследования:** управление «Умным домом» из одного центрального пункта.  
**Предмет исследования:** роботы из конструктора КЛИК

**Методы исследования:** формализация, обобщение, моделирование, эксперимент, описание.

**Для создания роботов были использованы:** КЛИК, стартовый набор 2.0–1 шт; ноутбук Леново – 1 шт; программа– mblock 5.

**1. Теоретическая часть**

**1.1.Программа Arduino**Arduino — это электронный конструктор и удобная платформа быстрой разработки электронных устройств для новичков и профессионалов. Платформа пользуется огромной популярностью во всем мире благодаря удобству и простоте языка программирования, а также открытой архитектуре и программному коду. Устройство программируется через USB без использования программаторов.

Arduino позволяет компьютеру выйти за рамки виртуального мира в физический и взаимодействовать с ним. Устройства на базе Arduino могут получать информацию об окружающей среде посредством различных датчиков, а также могут управлять различными исполнительными устройствами.

Микроконтроллер на плате программируется при помощи языка С++ и среды разработки Arduino. Проекты устройств, основанные на Arduino, могут работать самостоятельно, либо же взаимодействовать с программным обеспечением на mblokc 5. Платы могут быть собраны пользователем самостоятельно или куплены в сборе. Программное обеспечение доступно для бесплатного скачивания. Исходные чертежи схем (файлы CAD) являются общедоступными, пользователи могут применять их по своему усмотрению.

В 2006 Arduino получила признание в категории Digital Communities на фестивале Ars Electronica Prix .

Аппаратная часть платформы Arduino

Существует несколько версий платформ Arduino. Последняя версия Leonardo базируется на микроконтроллере ATmega32u4.

Uno, как и предыдущая версия Duemilanove построены на микроконтроллере Atmel ATmega328.

Старые версии платформы Diecimila и первая рабочая Duemilanoves были разработаны на основе Atmel ATmega168 ,более ранние версии использовали ATmega8.

Arduino Mega2560, в свою очередь, построена на микроконтроллере ATmega2560.

1. **Практическая часть** 
   1. **Этапы создания робота «Умный дом»**
2. Подготовительный этап.
3. Выбор темы проекта.
4. Обоснование его актуальности.
5. Постановка цели и задач проекта.
6. Составление плана работы по реализации проекта.
7. Обсуждение вариантов конструкции.
8. Выполнение проекта.
9. Конструкционный этап.
10. Присоединение элементов к основной микросхеме.
11. Дополнение проекта элементами для большей функциональности.
12. Внешнее оформление.
13. Программирование.
14. Написание программы.
15. Проверка совместимости программы со сборкой.

13. Изменение программы под конструкцию робота.

* 1. **Разработка устройства**

Проектирование системы можно разделить на 2 части: аппаратную и программную.

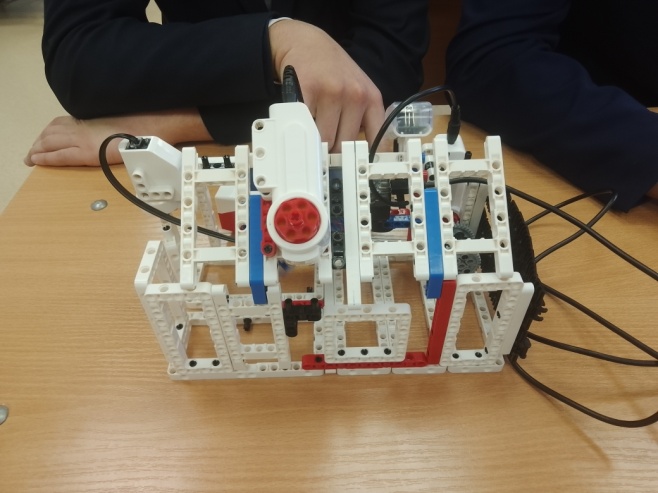
**Аппаратная часть**системы представлена платформой Arduino (рис.1) с расширениями и физическими модулями. Arduino – это инструмент для проектирования электронных устройств, более тесно взаимодействующих с окружающей физической средой, чем стандартные персональные компьютеры.

В данной работе будет использована версия платформы Arduino UNO – самая популярная версия базовой платформы Arduino с USB-интерфейсом и возможностью подключения большого разнообразия плат расширения. Эта платформа предназначена для физических расчетов (physical computing) с открытым программным кодом, построенная на простой печатной плате с современной средой для разработки программного обеспечения .

Arduino использует микроконтроллер Atmega328, который имеет 32 Кб флеш памяти. Этого будет вполне достаточно для выполнения возложенной на платформу задачи. Остальная обработка будет распределена на веб-ресурс.

Микроконтроллеры Arduino отличаются наличием предварительно прошитого в них загрузчика. С помощью этого загрузчика пользователь загружает свою программу в микроконтроллер без использования традиционных отдельных аппаратных программаторов и соединяется с компьютером через USB-интерфейс. Используя детали конструктора, собрали вентилятор, светофор, сушилку для рук и регулятор температуры. К ним прикрепили несколько датчиков.

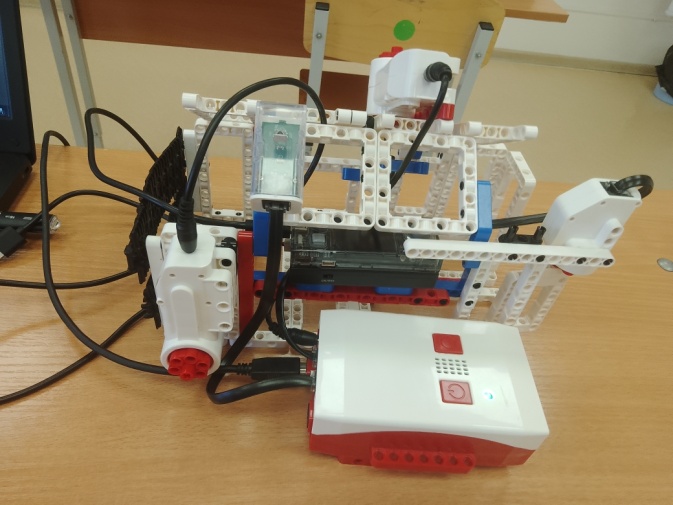
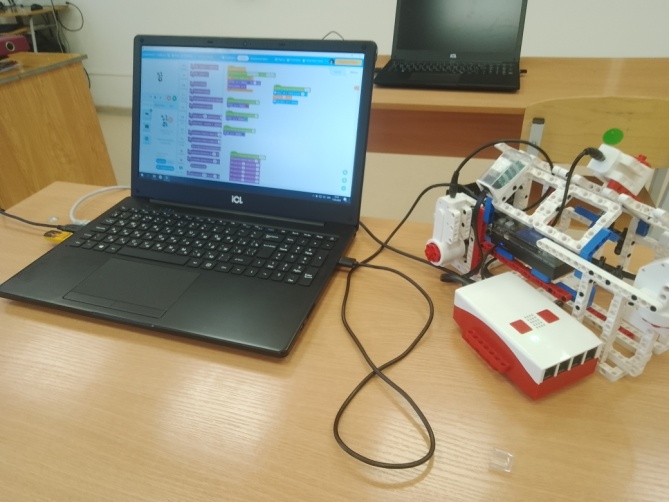
За освещение и мелодию будет отвечать cyberpi, а также за управление стиральной машиной (это условная модель имитирующая вращающийся барабан). За управление дверью, и вентилятором отвечает контроллер К1.

**Апробационный этап**

Нами были собраны и представлены для апробации: вентилятор, светофор, сушилка для рук и регулятор температуры. Вентилятор включается дистанционно.

В исходном состоянии должен гореть красный свет. После того, как пешеход нажмет кнопку I1, красный свет через три секунды сменится желтым, а еще через четыре секунды - зелёным. Зеленый свет горит 10 секунд, а потом снова загорается красный.  
Электросушилки для рук, оснащенные световыми барьерами для включения и отключения вентилятора.  
Датчик температуры непрерывно измеряет температуру воздуха. Как только температура в помещении превышает заданное максимальное значение, включается охлаждение воздуха. В другом случае, если температура снижается ниже установленного минимального значения, отключается охлаждение и включается нагрев воздуха.

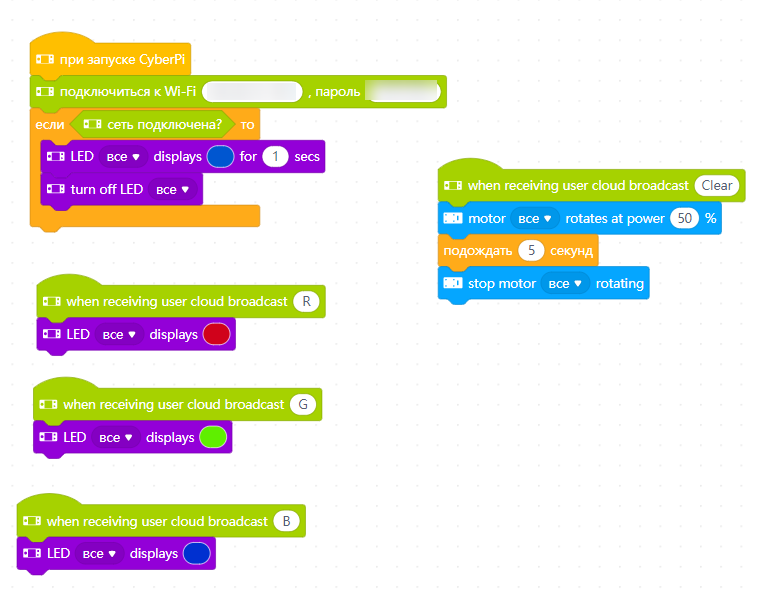
 

Для собранных роботов в программе mblokc 5 были составлены алгоритмы c ветвлением.

**Программная часть** системы представлена в программном коде микроконтроллера (Arduino скетч)

Напишем программу. Она будет состоять из трёх частей:  
- программа для управления через контроллер К1

- программа для управления через контроллер cyberpi

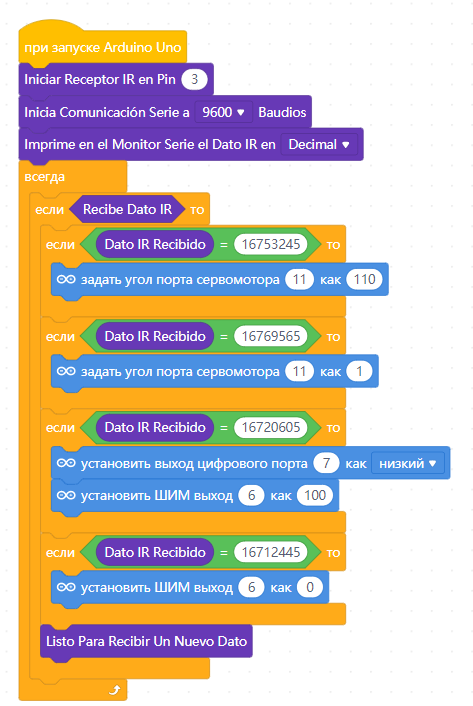


- программа для связывания работу двух контроллеров с событием на клавиатуре.

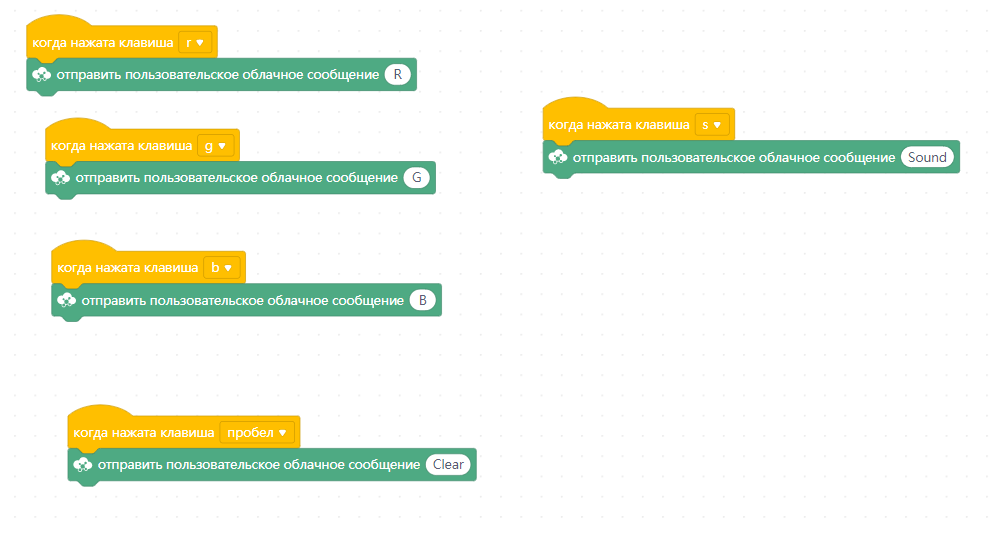


Программа для К1.

Управление с помощью IR модуля дверью и вентилятором.



Программа для персонажа (компьютера).  
Логика настройки взаимодействия клавиатуры с контроллером cyberpi



Также написание программы может быть через систему Arduine C. Данный код представлен в приложении 2

**Выводы и практические рекомендации**

Выводы и перспективы проекта: В результате работы над проектом мы пришли к выводу, что даже сложные проекты, постаравшись, может воплотить в жизнь любой человек, будь то взрослый или ребенок. Так, нам удалось создать робота, который способен включать вентилятор, свет и приборы, что значительно упрощает жизнь человека и его деятельность во многих сферах.

В будущем мы хотим усовершенствовать наш проект. Мы изучим дополнительные возможности робота и постараемся расширить его функционал, а также изучить особенности различных профессий и выявить сферы, деятельность в которых можно облегчить посредством использования данного робота.

**Заключение**

В результате работы над проектом нами созданы вентилятор, светофор, сушилку для рук и регулятор температуры для контроля и управления ими из одного центрального пункта.  
Поэтому можно сделать вывод о том, что цель проекта достигнута.

В этом проекте мы использовали не только микрокомпьютер и моторы, а мы постарались задействовать весь спектр имеющихся датчиков и сложных деталей из набора конструктора КЛИК Стартовый набор 2.0.

В программировании использовались многозадачность, условия, циклы. Программы написаны в среде программирования – mblock5.

Наша команда и далее будет продолжать работу над этим и другими проектами. Мы уверены, что наши идеи найдут свое применение для создания более комфортного и удобного дома, квартиры. Ведь создание удобства и уюта посредством контроля технических процессов — главное предназначение «Умного дома».

**Список литературы**

1. Григорьев Д. В., Степанов П. В. Внеурочная деятельность школьников: методический конструктор. – М.: Просвещение, 2020г. – 223с.
2. Копосов Д. Г. Первый шаг в робототехнику: практикум для 5–6 классов. - М.: БИНОМ. Лаборатория знаний, 2021г. – 286с.

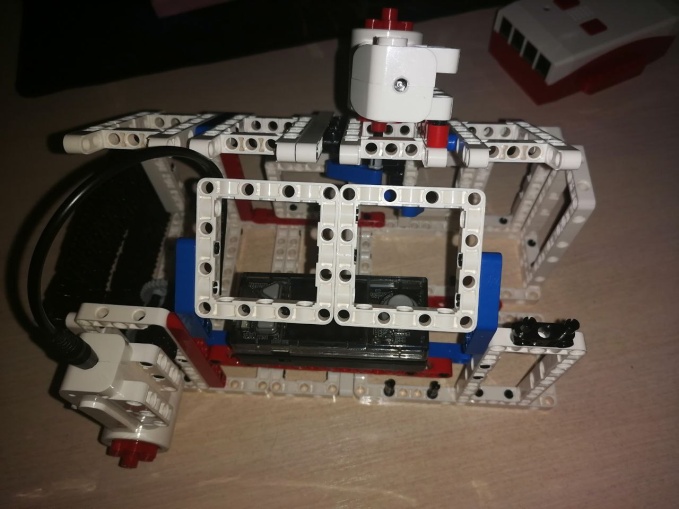
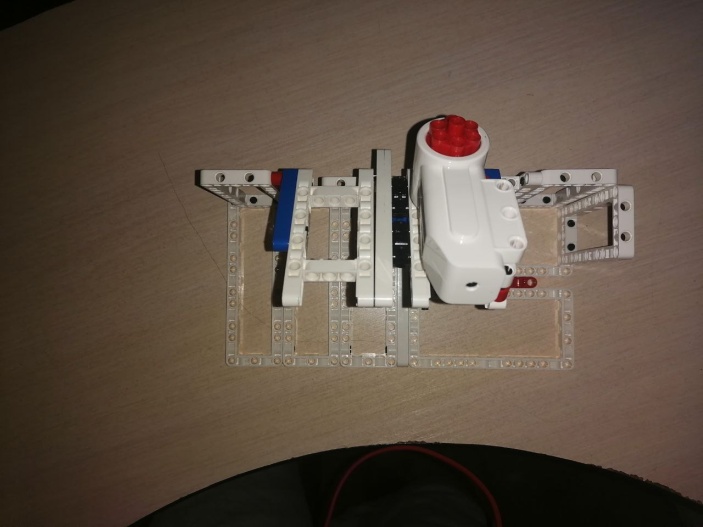
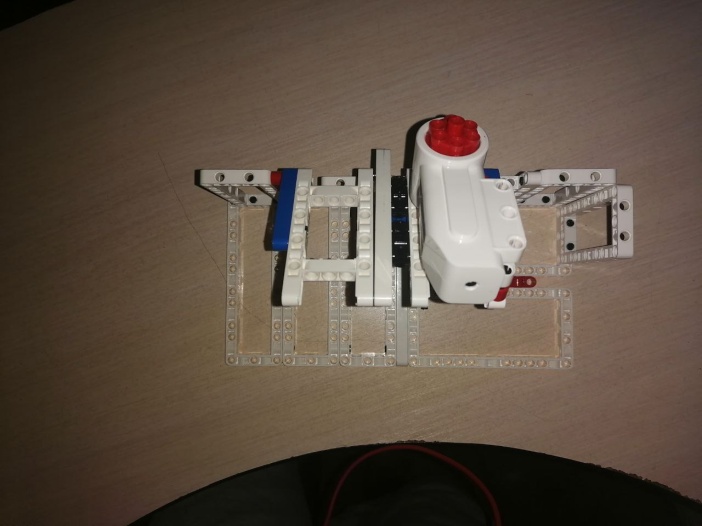
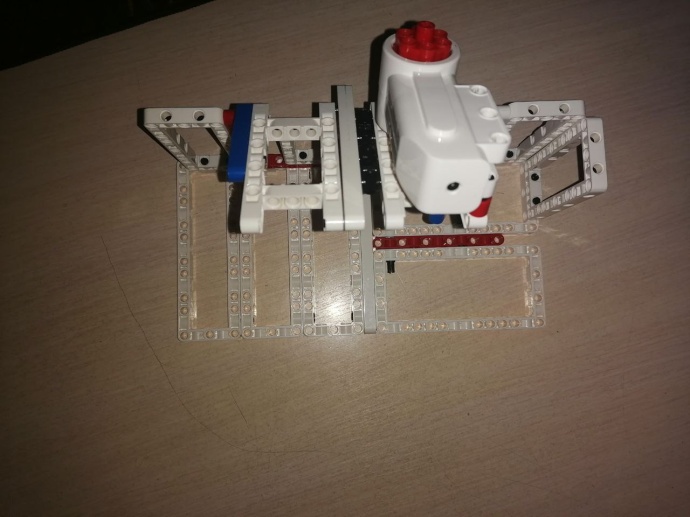
3. Юревич Е. И. Основы робототехники. – 2-е изд., перераб. и доп. – СПб.:БХВ-Петербург, 2020г. – 416с.

4. В. Архипов «Системы для «интеллектуального» здания» - "СтройМаркет", № 45 1999 г.

5. Mike Riley «Programming Your Home Automate with Arduino, Android, and Your Computer» - «The Pragmatic Bookshelf Dallas, Texas • Raleigh, North Carolina», 2012 г.

Интернет-ресурсы.

Приложение 1.

Приложение 2.

Код отвечающий за движение дома

// generated by mBlock5 for <your product>

// codes make you happy

#include <Servo.h>

#include <Arduino.h>

#include <Wire.h>

#include <SoftwareSerial.h>

Servo servo\_11;

void \_delay(float seconds) {

long endTime = millis() + seconds \* 1000;

while(millis() < endTime) \_loop();

}

void setup() {

servo\_11.attach(11);

pinMode(7,OUTPUT);

pinMode(6,OUTPUT);

// not implemented, yet

// not implemented, yet

// not implemented, yet

while(1) {

if(// not implemented, yet){

if(// not implemented, yet == 16753245.000000){

servo\_11.write(110);

}

if(// not implemented, yet == 16769565.000000){

servo\_11.write(1);

}

if(// not implemented, yet == 16720605.000000){

digitalWrite(7,0);

analogWrite(6,100);

}

if(// not implemented, yet == 16712445.000000){

analogWrite(6,0);

}

// not implemented, yet

}

\_loop();

}

}

void \_loop() {

}

void loop() {

\_loop();

}

Код CuberPi

# generated by mBlock5 for CyberPi

# codes make you happy

import event, time, cyberpi

# initialize variables

cyberpi.cloud\_broadcast.start('{TOPIC}')

@event.start

def on\_start():

cyberpi.wifi.connect('Keenetic-5196', 'FyRrPGkN')

if cyberpi.wifi.is\_connect():

cyberpi.led.on(0, 86, 208, "all")

time.sleep(1)

cyberpi.led.on(0, 0, 0, "all")

cyberpi.led.off("all")

@cyberpi.event.cloud\_broadcast('Clear')

def on\_cloud\_broadcast():

cyberpi.pocket.motor\_set(50,"all")

time.sleep(5)

cyberpi.pocket.motor\_stop("all")

@cyberpi.event.cloud\_broadcast('R')

def on\_cloud\_broadcast1():

cyberpi.led.on(208, 2, 27, "all")

@cyberpi.event.cloud\_broadcast('G')

def on\_cloud\_broadcast2():

cyberpi.led.on(95, 239, 0, "all")

@cyberpi.event.cloud\_broadcast('B')

def on\_cloud\_broadcast3():

cyberpi.led.on(0, 48, 208, "all")

@cyberpi.event.cloud\_broadcast('Sound')

def on\_cloud\_broadcast4():

cyberpi.audio.add\_tempo(10)

cyberpi.audio.play\_music(60, 0.4)

cyberpi.audio.play\_music(69, 0.4)

cyberpi.audio.play\_music(69, 0.4)

cyberpi.audio.play\_music(67, 0.4)

cyberpi.audio.play\_music(69, 0.4)

cyberpi.audio.play\_music(65, 0.4)

cyberpi.audio.play\_music(60, 0.4)

cyberpi.audio.play\_music(60, 0.4)

cyberpi.audio.play\_music(60, 0.4)

cyberpi.audio.play\_music(69, 0.4)

cyberpi.audio.play\_music(69, 0.4)

cyberpi.audio.play\_music(70, 0.4)

cyberpi.audio.play\_music(67, 0.6)

cyberpi.audio.play\_music(72, 0.2)

cyberpi.audio.play\_music(72, 0.4)

cyberpi.audio.play\_music(62, 0.4)

cyberpi.audio.play\_music(62, 0.4)

cyberpi.audio.play\_music(70, 0.4)

cyberpi.audio.play\_music(70, 0.4)

cyberpi.audio.play\_music(69, 0.4)

cyberpi.audio.play\_music(67, 0.4)

cyberpi.audio.play\_music(65, 0.4)

cyberpi.audio.play\_music(67, 0.4)

cyberpi.audio.play\_music(69, 0.4)

cyberpi.audio.play\_music(69, 0.4)

cyberpi.audio.play\_music(67, 0.4)

cyberpi.audio.play\_music(69, 0.4)

cyberpi.audio.play\_music(65, 0.6)