

Собираем игровую приставку «Pong» на Arduino Uno



- Платформы: Arduino Uno
- Языки программирования: Arduino (C++)
- Тэги: электронные игры, консоль, pong, dendy

Что это

Собираем простую игровую консоль на базе Arduino Uno. Подключаем плату к телевизору через композитный видеовход. Картинку генерируем с помощью библиотеки TVout, а игровую логику пишем на C++. В качестве контроллеров используем потенциометры — как и в раритетной Atari Pong.

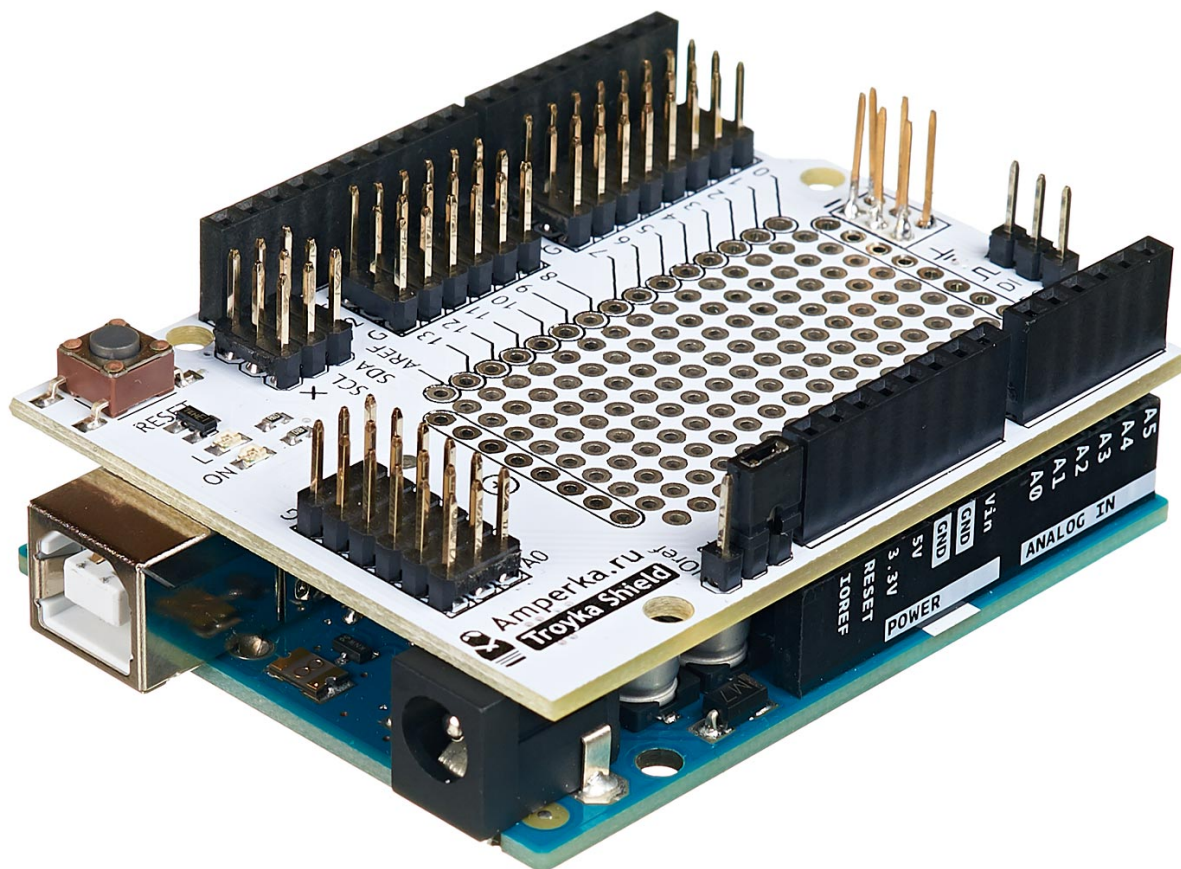
Что понадобится

1. Arduino Uno
2. Troyka Shield
3. Breadboard Mini
4. Потенциометр (Тройка-модуль) (2 шт.)
5. Ручка для потенциометра «Рекорд» (44 мм) (2 шт.)
6. Соединительные провода «папа-папа»
7. Герметичный корпус 120×120×60
8. Пластина мега (#Структор)
9. Кабель USB (A — B)
10. Импульсный блок питания (1000 мА)
11. Резистор 1 кОм (1 шт.)
12. Резистор 470 Ом (1 шт.)
13. Кабель соединительный 2×RCA «папа-папа» (тюльпаны)
14. Гнездо на панель RCA (2 шт.)
15. Гнездо питания на панель 2,1 мм
16. Герметичные разъёмы «папа-мама»
17. Многожильный монтажный провод с сечением 3×1 мм² (3 метра)

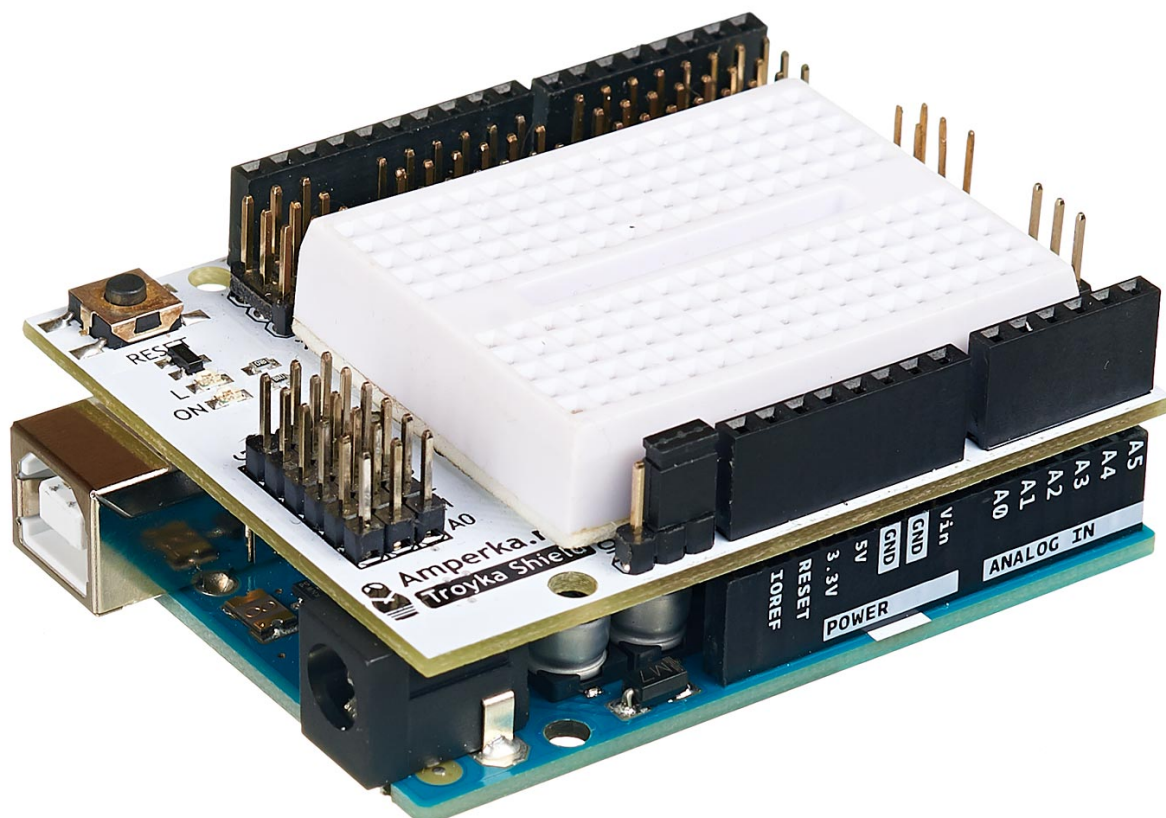
18. Кнопка сброса «звонок»

Как собрать

1. Установите Troyka Shield на плату Arduino Uno.

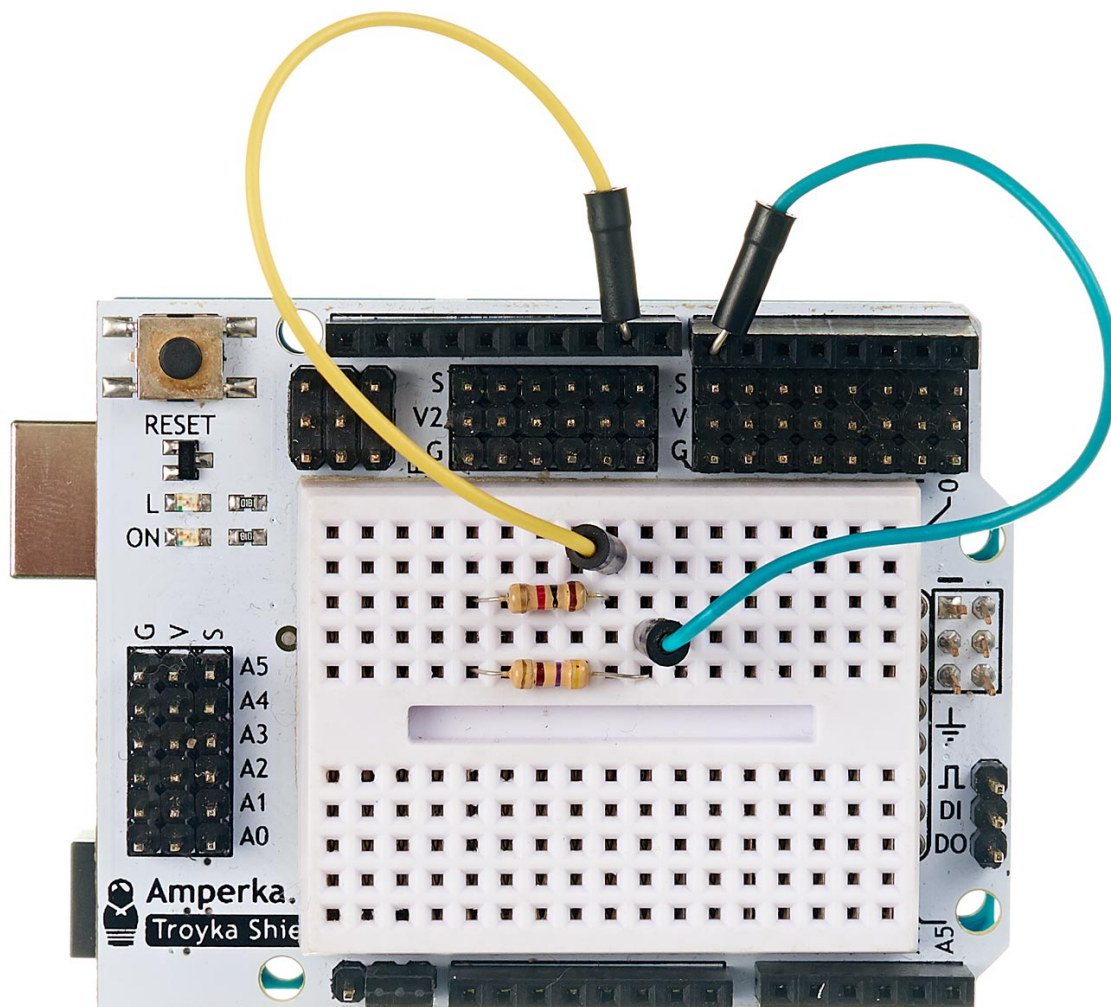


2. Приклейте макетную плату на Troyka Shield.



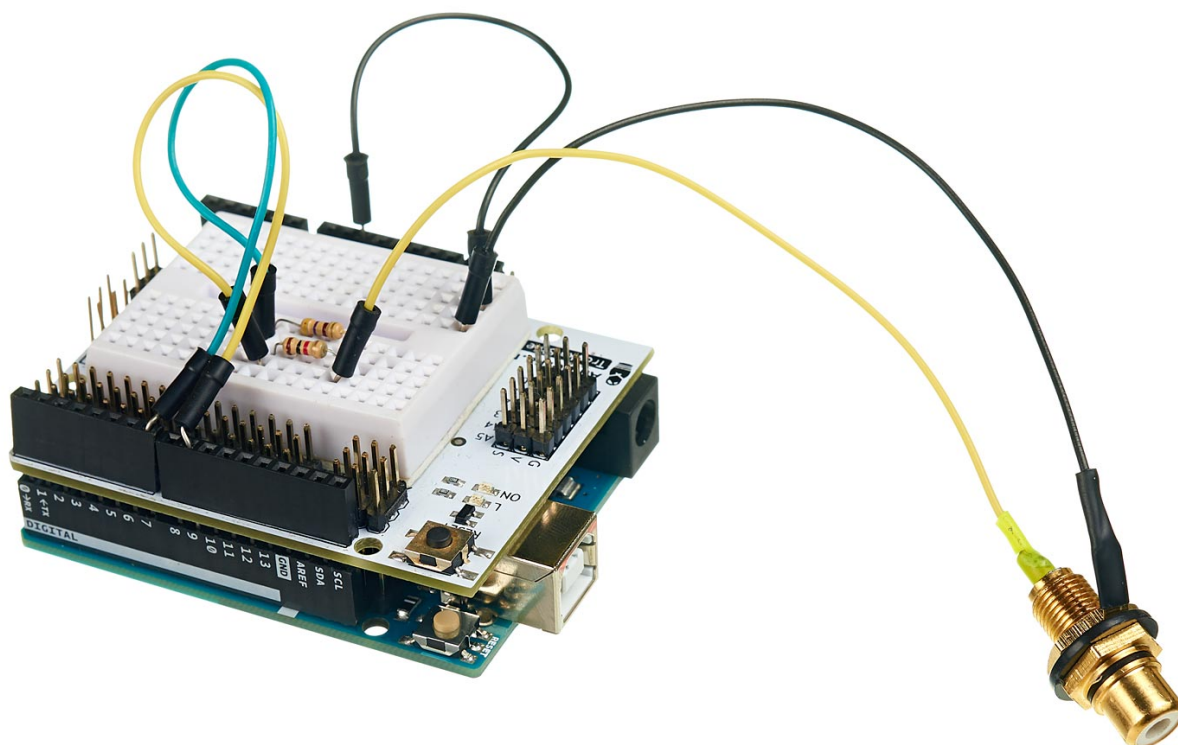
3. Вставьте в макетную плату два резистора на 470 Ом и 1 кОм и коммутируйте их следующим образом:
- Резистор 470 Ом соедините проводами «папа-папа» с 7 цифровым пином Arduino.
 - Резистор 1 кОм — с 9 цифровым пином Arduino.

- Соедините ножки резисторов резисторов между собой.

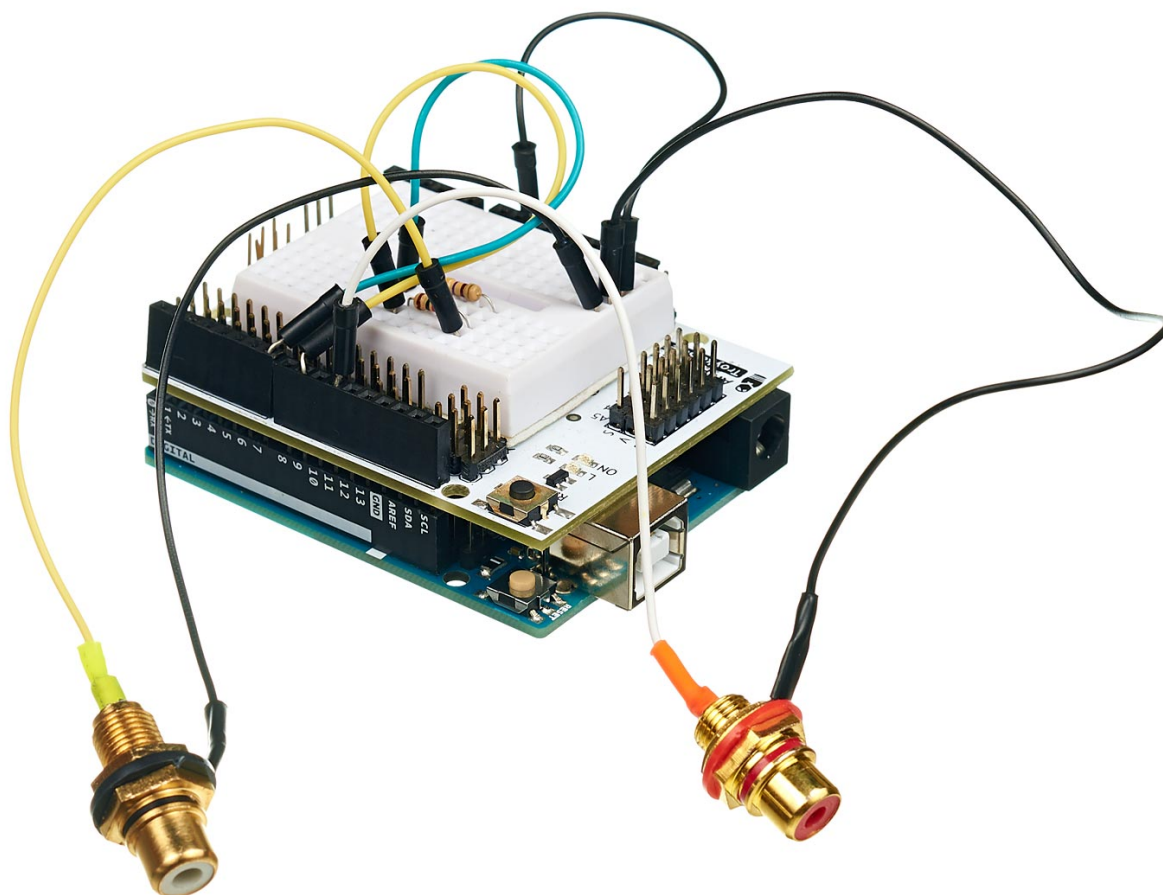


4. Возьмите RCA-разъём и подключите его центральный пин к точке соединения резисторов 470 Ом и 1 кОм. Внешний пин подключите к земле. Через этот разъём будем выводить

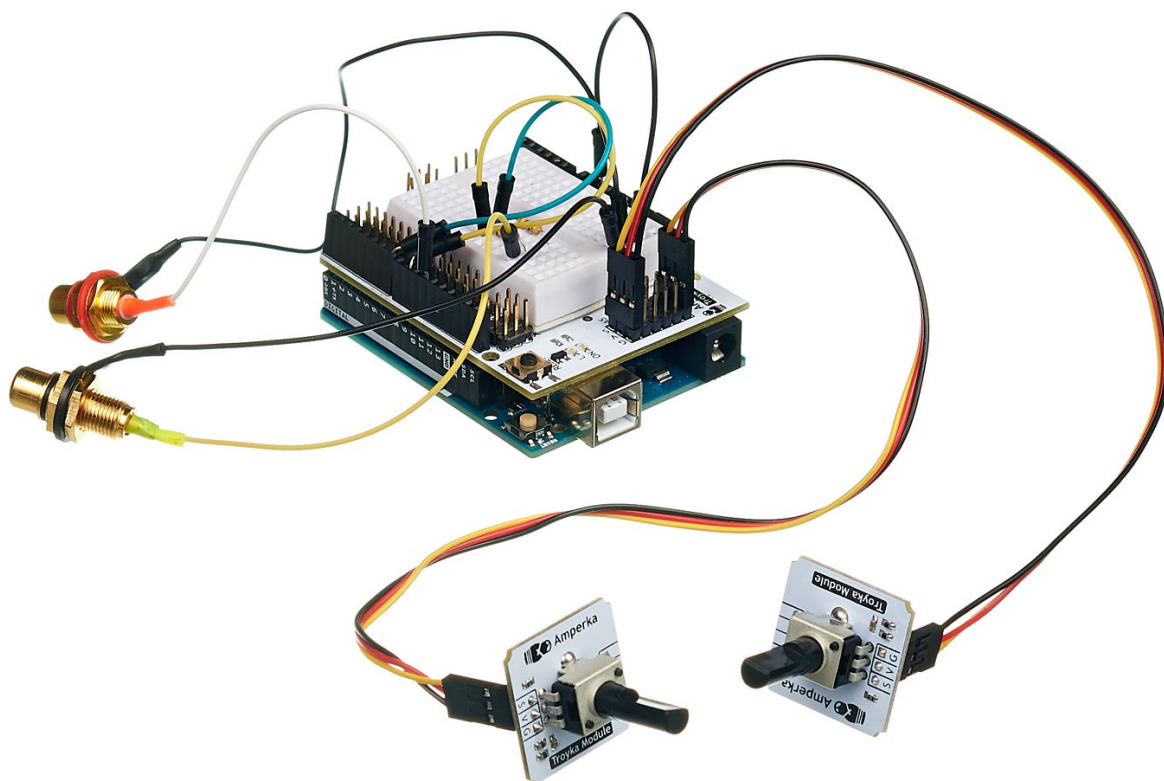
видеосигнал.



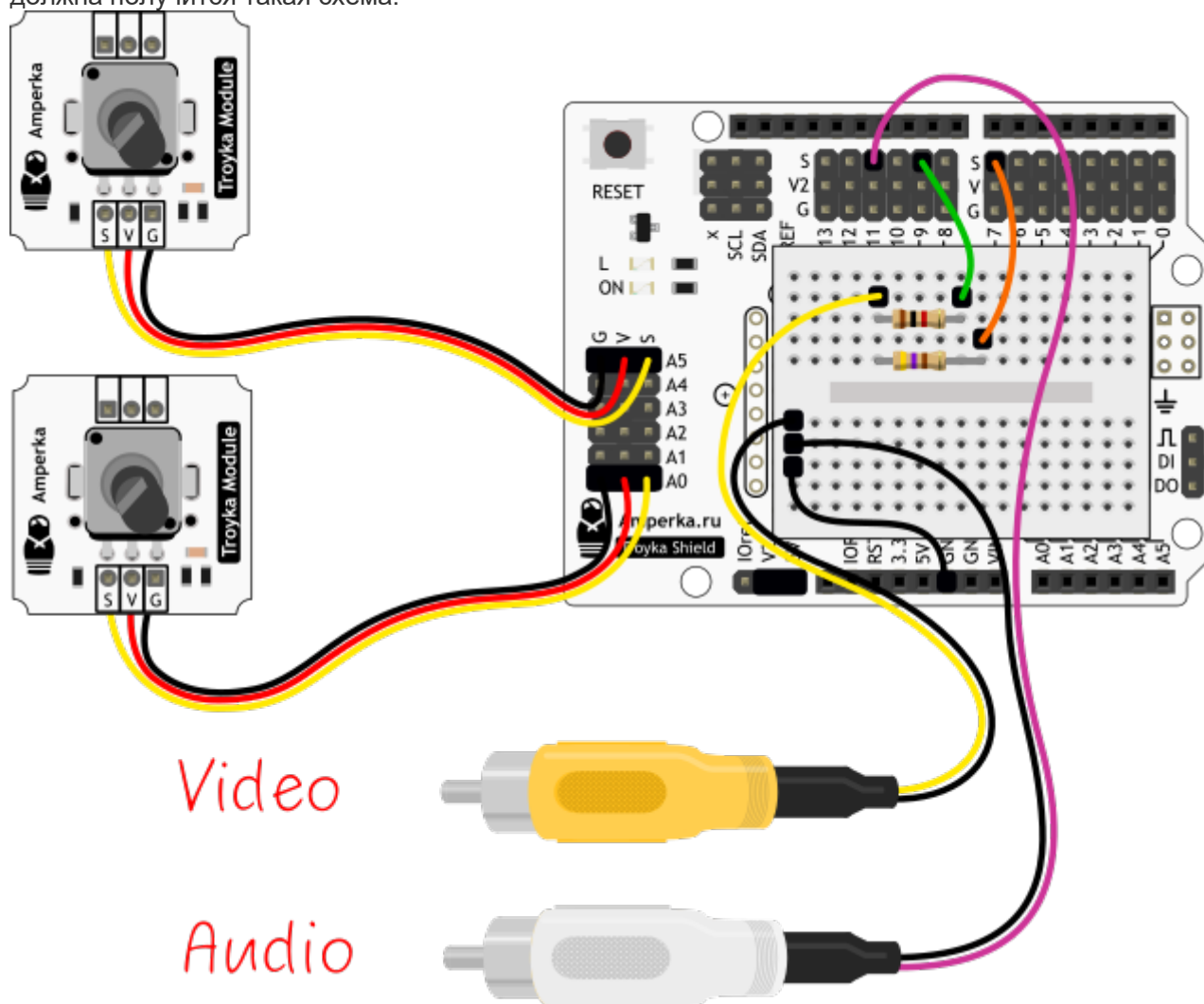
5. Возьмите второй RCA-разъём и подключите его к 11 пину Arduino и к земле. Этот разъём будет передавать аудиосигнал.



6. Подключите два потенциометра (Тройка-модуль) к Troyka Shield стандартными трёхпроводными шлейфами к аналоговым пинам A0 и A5.[611](#)

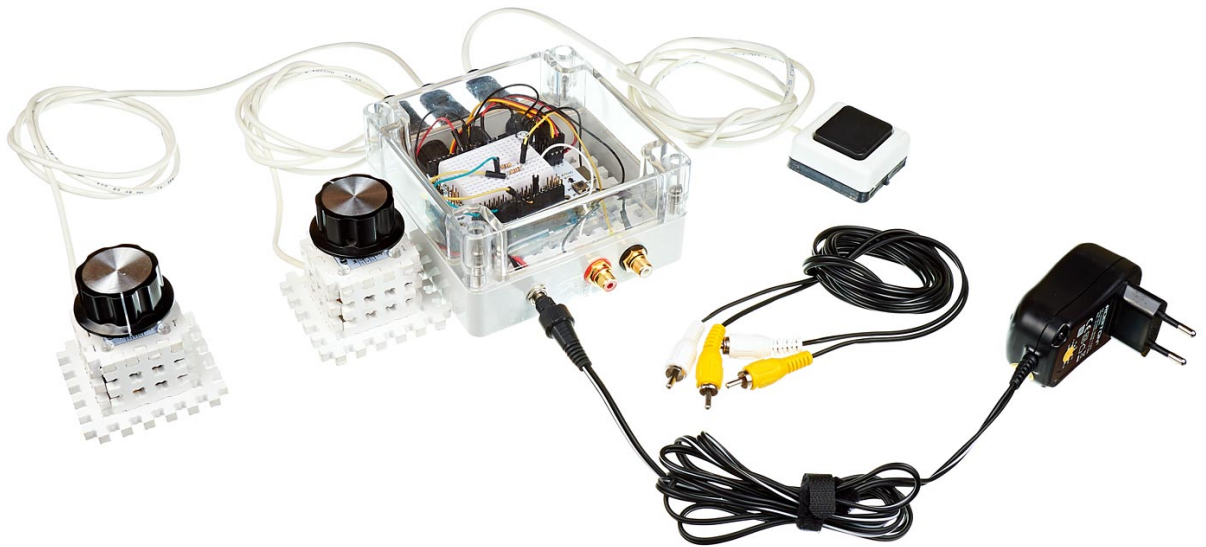


должна получиться такая схема:



- Для безопасности и красоты спрячьте управляющую электронику в корпус:

- Потенциометры упакуйте в #структор и подключите через герметичные разъёмы.
- Аналогично подключите кнопку старта игры. Используйте для этого пин `Reset`.
- Разъёмы аудио/видео и питание устройства разместите на противоположной панели корпуса.



Консоль готова. Подключите её соединительным кабелем 2×RCA «папа-папа» (тюльпаны) к телевизору, берите друга и наслаждайтесь игрой.

Исходный код

Для работы ниже приведённого скетча скачайте и установите библиотеку TVout для вывода информации в виде композитного видео сигнала.

[gamePong.ino](#)

```
// библиотека для работы с композитным видео выходом
#include <TVout.h>

// создаём объект TV класса TVout
TVout TV;

// максимальное количество очков
#define MAX_SCORE      7
// длина ракеток
#define PADDLE_HEIGHT  10
// невидимые боковые грани реккетки
// для отбивания в крайних точках ракетки
#define PADDLE_OFFSET  2

// пины подключения джойстиков каждого игрока
#define PLAYER_LEFT_PIN  A5
#define PLAYER_RIGHT_PIN A0

// переменные для хранения размеров экрана
int hres, vres;

// координаты шара
```

```

int ballX, ballY;
// направление шара
int ballDX = 1;
int ballDY = 1;
// очки игроков
int playerScoreLeft = 0;
int playerScoreRight = 0;
int leftPaddleY = 0;
int rightPaddleY = 0;

// СОСТОЯНИЯ СИСТЕМЫ
enum State
{
    RESET_GAME,
    NEXT_LEVEL,
    PLAY_GAME,
    STATE_MISS,
};

// объявляем переменную state
State state;

bool missed = 0;

void setup()
{
    // инициализируем коммуникацию с телевизором
    TV.begin(NTSC, 136, 96);
    // ждём 1 секунду
    delay(1000);
    // считываем размеры экрана
    hres = TV.hres();
    vres = TV.vres();
    // сбрасываем игру
    state = RESET_GAME;
}

void loop()
{
    switch (state) {
        case RESET_GAME:
            // очищаем экран
            TV.clearScreen();
            // выбираем шрифт «4×6»
            TV.selectFont(font4x6);
            // печатем на экране слово «Амперка»
            TV.print(55, 0, "Amperka");
            delay(1000);

```



```

        // выбираем шрифт «8×8»
        TV.selectFont(font8x8);
        // печатем на экране название игры «Arduino Pong»
        TV.print(20, 30, "Arduino Pong");
        delay(1000);
        // на старт, внимание, вперёд
        for (int i = 3; i != 0; i-- ) {
            TV.print(hres / 2, 60, i);
            TV.tone(1000, 300);
            delay(1000);
        }
        TV.tone(2000, 300);
        // выбираем шрифт «4×6»
        TV.selectFont(font4x6);
        // очищаем экран
        TV.clearScreen();
        // обнуляем счёт обоих игроков
        playerScoreLeft = 0;
        playerScoreRight = 0;
        // переходим на следующий уровень
        state = NEXT_LEVEL;
        break;
    case NEXT_LEVEL:
        // сброс шарика и ракеток
        resetBallAndPaddles();
        // рисуем игровое поле на экране
        drawBox();
        // выводим очки игроков на экране
        drawScores();
        // рисуем игровые ракетки
        drawPaddles();
        // переходим в состояние игры
        state = PLAY_GAME;
        break;
    case PLAY_GAME:
        // если мяч достиг верхней / нижней границы поля
        if (ballY == vres || ballY == 0) {
            ballDY *= -1;
        }
        // если мяч приближается к правой стороне
        if (ballX >= hres - 2) {
            // если мяч отбился правой ракеткой
            if (ballY > rightPaddleY - PADDLE_OFFSET &&
                ballY < (rightPaddleY + PADDLE_HEIGHT + PADDLE_OFFSET) && ballDX > 0
            ) {
                ballDX = -1;
            }
        }
        // если мяч достиг правой стены
        if (ballX == hres - 1) {

```

```

        missed = true;
        state = STATE_MISS;
        playerScoreLeft++;
        break;
    }

    // если мяч приближается к правой стороне
    if (ballX <= 2) {
        // если мяч отбился левой ракеткой
        if (ballY > leftPaddleY - PADDLE_OFFSET &&
            ballY < (leftPaddleY + PADDLE_HEIGHT + PADDLE_OFFSET) && ballDX < 0 )
        {
            ballDX = 1;
        }
    }

    // если мяч достиг левой стены
    if (ballX == 0) {
        missed = false;
        state = STATE_MISS;
        playerScoreRight++;
        break;
    }

    // обновляем положения ракеток
    leftPaddleY = map(analogRead(PAYER_LEFT_PIN), 0, 1024, 0, vres -
PADDLE_HEIGHT);
    rightPaddleY = map(analogRead(PAYER_RIGHT_PIN), 0, 1024, 0, vres -
PADDLE_HEIGHT);
    // обновляем положение шарика
    drawBall();
    // меняю местоположение шарика на следующий шаг
    ballX += ballDX;
    ballY += ballDY;
    drawPaddles();
    // обновляем положение шарика
    drawBall();
    TV.delayFrame(1);
    break;

case STATE_MISS:
    // если кто то пропустил мяч
    if (playerScoreLeft == MAX_SCORE) {
        TV.print(16, vres / 2, "Winner!");
        TV.tone(2000, 500);
        TV.delayFrame(120);
        state = RESET_GAME;
        while(1);
        break;
    } else if (playerScoreRight == MAX_SCORE) {
        TV.print(hres / 2 + 16, vres / 2, "Winner!");
    }
}

```

```

        TV.tone(2000, 500);
        TV.delayFrame(120);
        state = RESET_GAME;
        while(1);
        break;
    }
    if (missed) {
        TV.print(hres / 2 + 16, vres / 2, "Missed!");
        TV.tone(500, 300);
    } else {
        TV.print(16, vres / 2, "Missed!");
        TV.tone(500, 300);
    }
    delay(1000);
    TV.clearScreen();
    state = NEXT_LEVEL;
    break;
}
}

// сброс шарика и ракеток
void resetBallAndPaddles() {
    randomSeed(analogRead(A2));
    int noise = random(vres);
    ballX = hres / 2;
    ballY = random(vres);
    ballDX = (noise & 0x01) ? 1 : -1;
    ballDY = (noise & 0x02) ? -1 : 1;

    leftPaddleY = vres / 2;
    rightPaddleY = vres / 2;
}

// вывод очков на дисплей
void drawScores() {
    TV.printChar((hres / 4), 4, '0' + playerScoreLeft);
    TV.printChar((hres / 4) + (hres / 2), 4, '0' + playerScoreRight);
}

// вывод ракеток на дисплей
void drawPaddles() {
    // стираем предыдущие ракетки
    // стираем всю возможную область их местоположения
    TV.drawRect(0, 0, 1, vres, 0, 0);
    TV.drawRect(hres - 2, 0, 1, vres, 0, 0);

    // выводим текущее положение ракеток
    TV.drawRect(0, leftPaddleY, 1, PADDLE_HEIGHT, WHITE, true);
    TV.drawRect(hres - 2, rightPaddleY, 1, PADDLE_HEIGHT, WHITE, true);
}

```

```
// вывод шарика
void drawBall() {
    // рисуем шарик
    TV.setPixel(ballX, ballY, INVERT);
}

// отрисовка игрового поля на экране
void drawBox() {
    TV.clearScreen();
    // середина поля
    for (int i = 0; i < vres; i += 6) {
        TV.drawLine(hres / 2, i, hres / 2, i + 2, 1);
    }
    // верхняя граница поля
    TV.drawLine(0, 0, hres, 0, WHITE);
    // нижняя граница поля
    TV.drawLine(0, vres - 2, hres, vres - 2, WHITE);
}
```