

Физтех-фабрика

Копилка с автоматическим счётчиком монет

Выполнили: Бочкарев Федор, Б01-003
Бочкарева Софья, Б01-003

1 Цель

Построить копилку с автоматическим определением номинала монет и накоплением общей суммы.

Задачи

- Анализ существующих решений;
- Проектирование и изготовление устройства;
- Проверка и верификация надёжности работы

2 Общее описание

Копилка со счетчиком монет. В монетной щели расположена клиновидная задвижка, соединенная к шаговому мотору и фототранзистору. С помощью фототранзистора фиксируется наличие/отсутствие монеты в щели, а ее номинал определяется по повороту шагового мотора.

Данные о принятых монетах выводятся на дисплей в двух режимах: общая сумма в копилке и распределение содержимого по номиналам.

3 Анализ существующих аналогов

Копилки со счетчиками монет в различных вариациях представлены на маркетплейсах [1], [2], однако такие устройства либо не являются автоматическими, и требуют ручного ввода добавленной суммы; либо их цена сильно завышена[3].

Определять номинал монеты можно по одному из трех признаков: диаметр, толщина или состав (магнитные свойства). Определение номинала по магнитным свойствам возможно только с очень чувствительными (дорогими) датчиками. Определение номинала по толщине требует высокой точности – до 0,01 мм. Разница в диаметрах монет составляет минимум 1 мм. Поэтому реализовать механизм определения номинала монеты целесообразнее именно по диаметру.

Пример устройства, реализующего такой механизм, выполнен AlexGyver [4]. В данной идее диаметр определялся по интенсивности инфракрасного сигнала, приходящего на фоточувствительный элемент (фототранзистор). Такая реализация допускает возможность ошибки за счет проскакивания монеты и неправильного считывания ее диаметра. К тому же в этой версии не предусмотрены удобные элементы управления (кнопки reset/переключение дисплея).

Другими аналогичными устройствами являются многочисленные реализации сортировщиков монет, основанные на проскакивании монет через щели определенного диаметра [5]. Такие устройства обладают высокой скоростью сортировки и счета, однако механизм занимает достаточно много места. Эта реализация хорошо подходит для промышленного использования (большого оборота монет), поэтому существует широкий сегмент подобных устройств, обладающих высокой надежностью и, соответственно, высокой ценой.

Таким образом, наше устройство выделяется на фоне существующих непромышленных аналогов автоматической и надежной работой, что делает его удобным в использовании.

4 Проектирование и изготовление

Основные этапы работы:

- Проектирование и прототипирование;
- Изготовление деталей;
- Сборка;
- Разработка управляющей программы

4.1 Проектирование и прототипирование

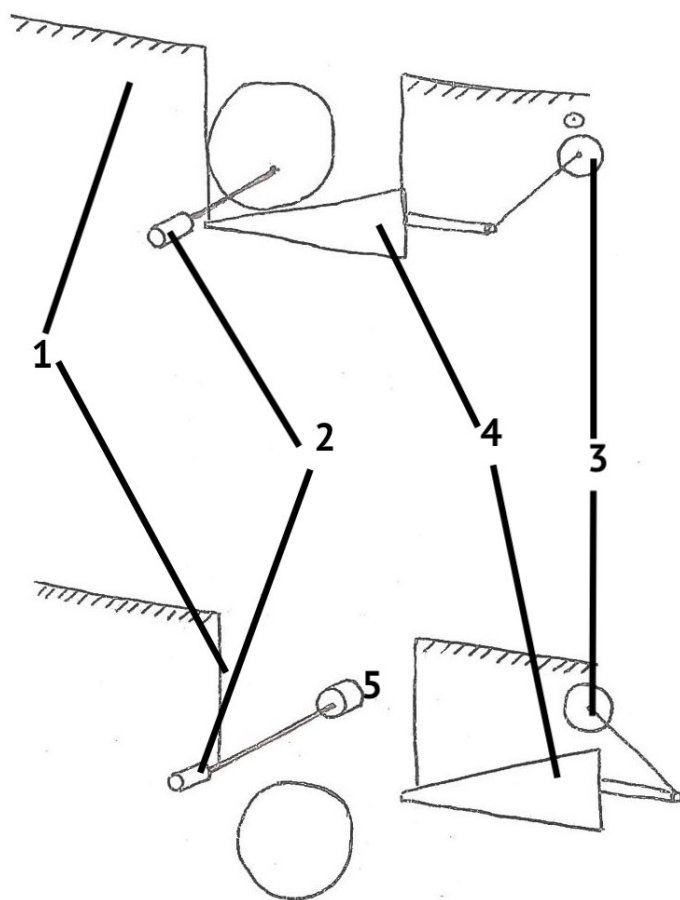


Рис. 1: Принципиальная схема устройства

Принципиальная схема устройства приведена на рис. ???. Монета падает в щель монетоприемника 1 и перекрывает инфракрасный луч от инфракрасного светодиода 2. Фоторанзистор 5 передает управляющий сигнал на шаговый мотор 3, который приводит в движение клин 4. Когда клин отводится достаточно для того, чтобы монета упала насквозь, инфракрасный луч светодиода 2 перестает перекрываться, что фиксируется фототранзистором.

По данной принципиальной схеме был собран прототип (см. рис. ??). Он успешно прошел проверку на работоспособность, что позволило нам использовать данный механизм в дальнейшем.

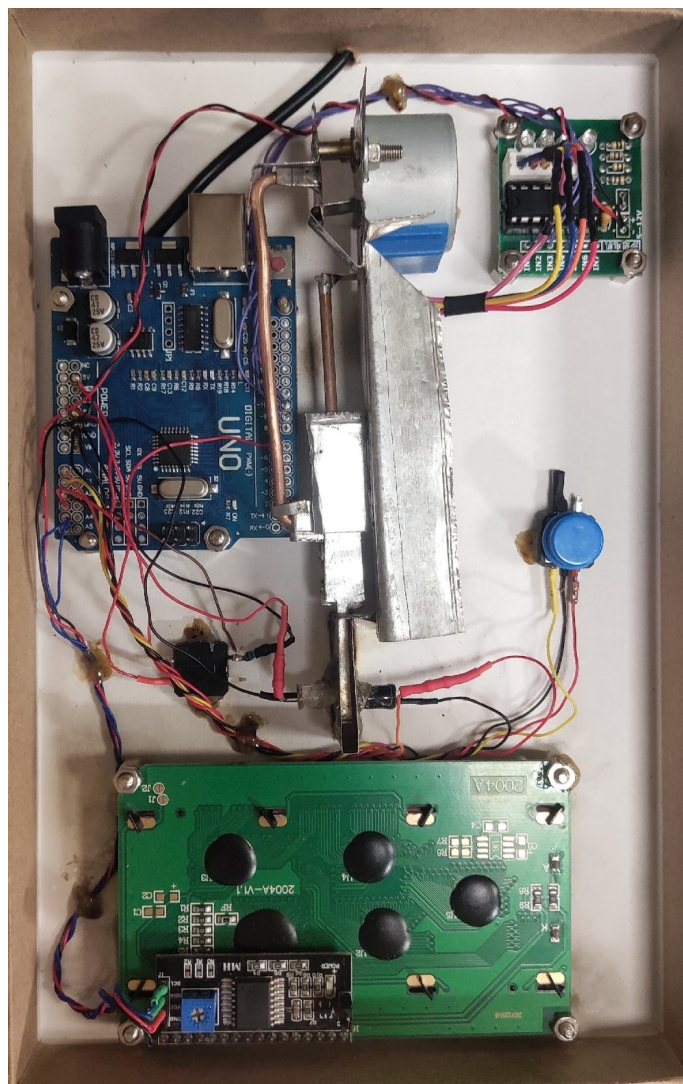


Рис. 2: Прототип

При проектировании финальной версии мы доработали механизм затвора для его длины (со 120 мм до 60 мм) (см. рис. ??). Корпус копилки состоит из двух основных составных частей - крышки, на которой закреплены все основные детали механизма, и коробки для монет.

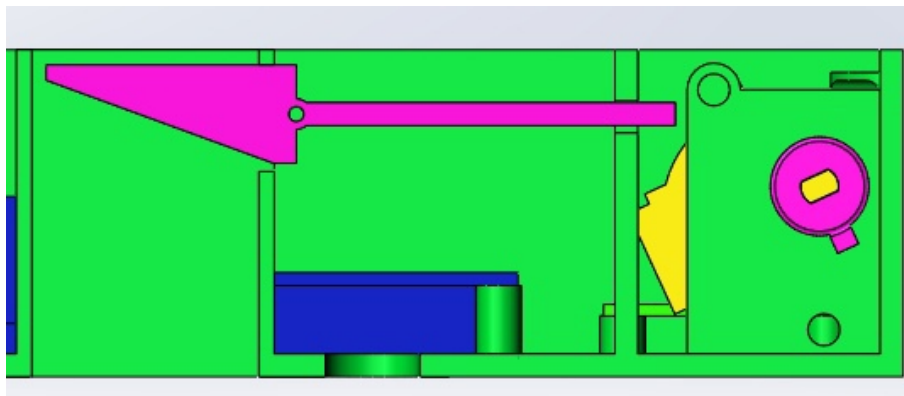


Рис. 3: Реализация механизма

На верхней крышке предусмотрено размещение 2-х кнопок, дисплея, модуля Arduino Nano,

драйвера шагового мотора и его драйвера, фототранзистора и инфракрасного светодиода и механизма затвора. Питание осуществляется от постоянного тока 5V от сети (см. рис. ??, рис. ??).

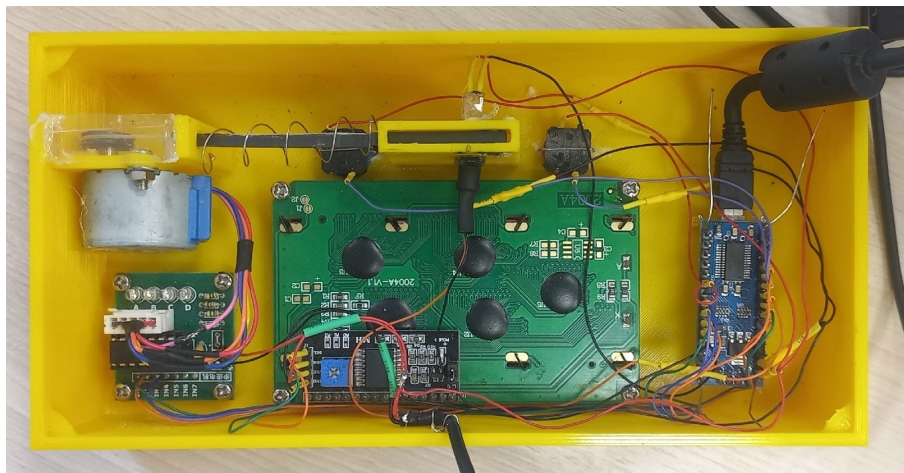


Рис. 4: Крышка – реализация

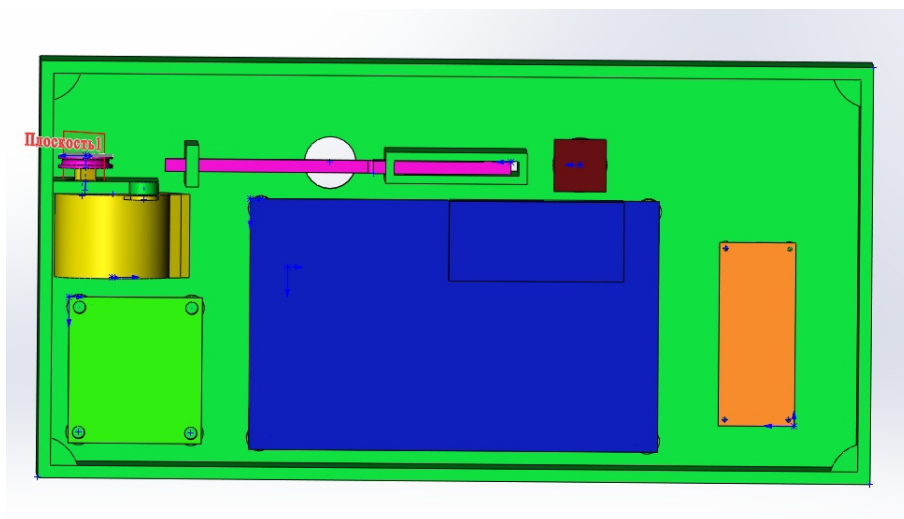


Рис. 5: Крышка – модель

4.2 Изготовление деталей

Все детали были изготовлены на 3д принтере. Изначально коробка была сделана без ребер жесткости, из-за чего при печати она деформировалась. Поэтому мы изменили конструкцию коробки, добавив ребра жесткости.

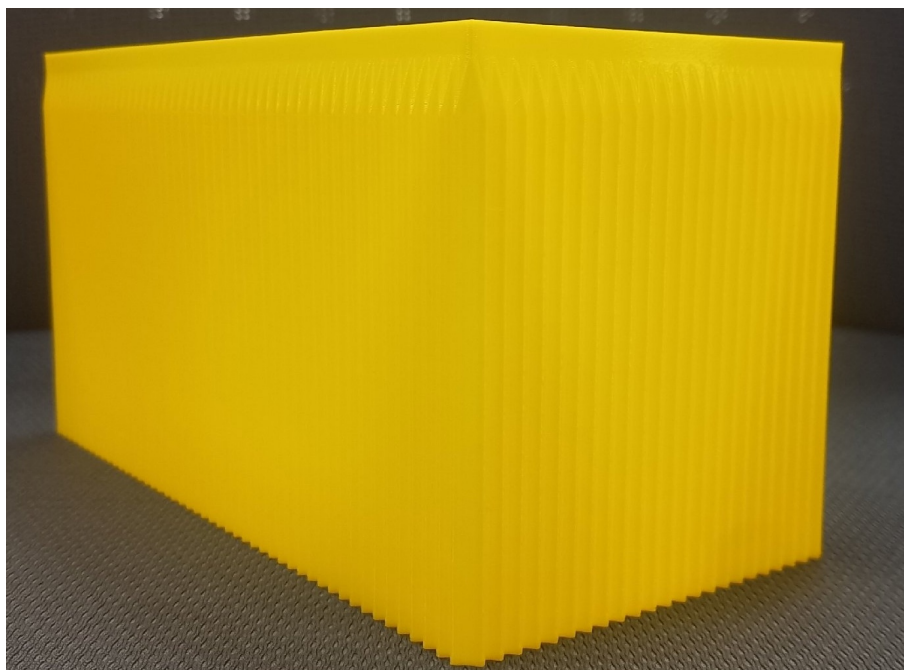


Рис. 6: Короб для монет

4.3 Сборка

Основным этапом стала сборка итогового устройства.

Электрическая схема была собрана из: шагового мотора, платы-драйвера шагового мотора, Arduino Nano, фототранзистора, ИК-светодиода, LCD экрана формата 2004 вместе с платой I2C, 2-х кнопок и провода USB (для питания). Схема соединений приведена ниже на рисунке ??.

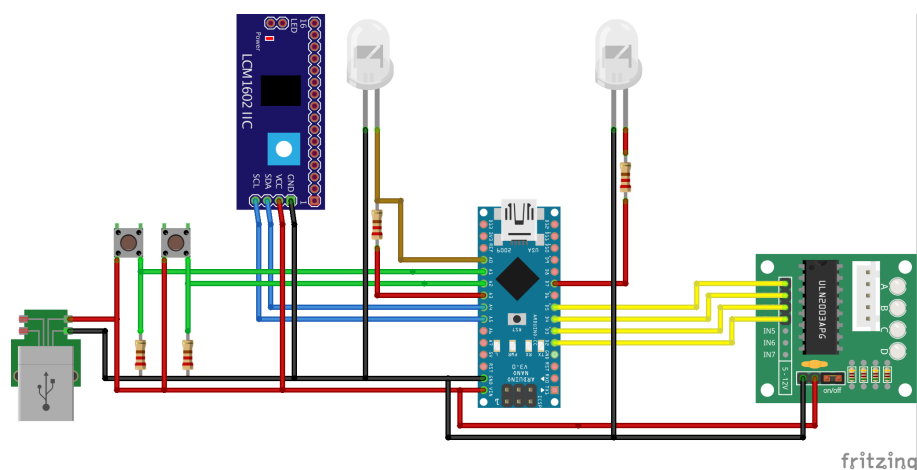


Рис. 7: Схема соединения проводов

При сборке были исправлены выявленные ошибки моделирования. В основном это касается отсутствующего отверстия для щели под ИК-светодиод и фототранзистор.

В ходе процесса калибровки открытия щели для определенного номинала монет, была многократно доработана система задвижки – методом перебора были отсеяны 2 наиболее подходящие пружины из порядка 20 оставшихся и одна из них была растянута/раскручена для достижения надёжной работы механизма.

С ходе испытаний, шаговый мотор смог повернуть стопорное положение (упор в корпус) и тем самым деформировал стенку корпуса. Для восстановления работоспособности механизма задвижки корпус крышки был незначительно деформирован – была вставлена насквозь в стенку в месте упора небольшой штырёк, обеспечивающий надёжное стопорное положение для шагового мотора.

4.4 Разработка управляющей программы

Основу прошивки Arduino Nano составляют добавленные библиотеки.

```
1 #include <GyverStepper.h>
2 #include <EEPROM.h>
3 #include <PinChangeInterrupt.h>
4 #include <string.h>
5 #include <LiquidCrystal_I2C.h>
```

В дальнейшем, весь код можно разделить на части:

- Объявление констант, классов, переменных и дефайнов;
- Основная работа;
- Функции/код связанные с экраном и его выводом;
- Функции связанные с записью количества монет в энергонезависимую EEPROM память;
- Обработка клавиш управления

Вся основная работа заключена в простой алгоритм: постоянно считываются данные с сенсора (фототранзистора), если они превысили определенный порог (сенсор заслонили от ИК), то у нас монетка и нужно отодвигать защёлку путём поворота шагового мотора. Для каждой монетки существует соответствующий угол, который напрямую связан с диаметром монеты. Зная на каком углу датчик вновь увидел ИК мы можем определить какая это монета.

```
1 void loop {
2     data = analogRead();
3     if (data > BORDER) {
4         if (check(...)) { this_coin = 0 }
5         else if (check(...)) { this_coin = 1 }
6         ...
7     }
8 }
```

Из вышеприведённой вставки, главная функция имеет следующий функционал: поворачивается на определённый угол R , затем даёт время на возможный сброс монетки и срабатывание датчика Y , потом вновь считывает датчик и определяет – упала ли монета или нет.

```
1 bool check(int * mass, int R, int Y){
2     int data = analogRead(inSEN);
3     if (data > 100) {
4         my_rotate(R);
```

```

5      } else {
6          comeback ();
7          return 1;
8      }
9      delay (Y);
10     return 0;
11 }

```

Если монета была сброшена, то эта функция прорутит мотор до стопора, тем самым вернув защёлку в полностью закрытое состояние.

По результату выхода из этой функции определяется тип монетки и он записывается в буфер имеющихся монет.

По данным этого буфера на экран выводиться сумма и количество монет по номиналам.

5 Тестирование

Основным этапом процесса тестирования стала калибровка шагового мотора для его надежного срабатывания на каждый номинал монеты.

В процессе тестирования были выявлены недостатки модели, как то: отсутствие в щели монетоприемника отверстия для датчика, слишком большая жесткость выбранной пружины (проявляется при больших отклонениях мотора).

6 Результаты

Было спроектировано, изготовлено и протестировано устройство — копилка со счетчиком монет. Функционально устройство работает исправно.

В качестве дальнейшего развития можно доработать управляющую программу и внедрить в нее новые функции, а также доработать модель и исправить недостатки, выявленные при тестировании модели.