­­ Добрый день. Рад приветствовать слушателей и членов жюри. ­­Я хочу представить вам свой проект на тему «СОЗДАНИЕ АВТОМАТИЧЕСКОЙ СИСТЕМЫ ПОДАЧИ ЗВОНКОВ». В двадцать первом веке организация времени является чрезвычайно важным умением, которое во многом облегчает жизнь и ее планирование. Во многих образовательных учреждениях школьные звонки подаются вахтером вручную. Моя родная школа, 64 лицей – не исключение. Очевидно, что у этого метода есть недостатки: из-за человеческого фактора звонок всегда дается раньше или позже, на пару секунд, или на минуту. В случае применения автоматизирующего устройства будет освобожден человек, который сможет выполнить сложные задачи.

**Hardware**

Разработка началась с исследования. В ходе моего наблюдения было выявлено, что для интеграции будущей системы необходимо подключиться, например, параллельно к выключателю, чтобы оставить возможность ручной подачи (в случае непредвиденных поломок). Целью моего проекта является создание достаточно простой, но в то же время удобной и доступной системы. Обязательным условием было включение дисплея в готовое устройство. Именно он позволит исключить необходимость в специалисте для настройки системы. Также, это наглядная иллюстрация происходящего. Готовую схему вы можете посмотреть на слайде презентации. Задействован модуль реального времени DS3231, энкодер, OLED дисплей 128 на 64 пискеля, тумблер, а также микроконтроллер Arduino. О нем и стоит поговорить. Из множества плат я выбрал три наиболее подходящих. Первая – Arduino Nano, в основе которой лежит ATmega 168. (16 Кб флеш-памяти, 1 Кб ОЗУ). Вторая – Arduino Nano на ATmega328p (32 Кб flash и 2 Кб ОЗУ). Третья плата – самая серьезная, Arduino Mega на ATmega 2560 (256 Кб флеш-памяти и 8 Кб ОЗУ) Первые две идентичны по цене, а Mega в 3 раза дороже. Пришлось пожертвовать вычислительной мощностью, чтобы удешевить проект. Тем более, итоговая прошивка заняла около 70 процентов динамической памяти. В самом конце сборки я решил предусмотреть возможность установки обычного переключателя, чтобы подать звонки вручную. Также, достаточно вероятна замена электромагнитного реле на твердотельное. В первом случае замыкание происходит механически: раздается щелчок, изнашивается механизм. Твердотельное реле работает бесшумно и не изнашивается.

**Software**

«Железо» выбрано, теперь нужно было заставить всё вместе работать. В качестве окружения для разработки идеально подходит Arduino IDE. На презентации вы можете увидеть QR код, который перенаправит вас на ресурс GitHub: там хранятся все исходники. Глобальный отсчет времени ведется в секундах, преобразуется из часов, минут и секунд. Само меню реализовано на нескольких функциях, которые вызываются методом switch-case, он заменил огромное количество логических операторов если-то. Для обработки ввода с энкодера были использованы стандартные аппаратные прерывания (attachInterrupt ()). Эта функция позволяет во время выполнения основной программы словно бросить все и обработать нажатие или поворот энкодера. Если пренебречь таким решением, то ввод будет обрабатываться очень некорректно, управление становится невозможным. Данная проблема действительно возникла в процессе разработки, отняла целый день работы. В свою очередь, логика подачи самого сигнала достаточно проста: текущее значение сравнивается со значениями таймингов звонка и подает сигнал на управляющий пин реле, если они совпадают.

Элементы управления:

1. поворот право-лево навигация по пунктам
2. нажатие в главном меню: выбор пункта
3. нажатие или удержание в подпунктах меню: возвращение назад
4. поворот с удержанием в меню настроек: настройки времени

На слайде вы можете увидеть фотографии всех пунктов меню.

Достаточно интересный пункт – «сон». Многим известно, что OLED дисплеи подвержены выгоранию пикселей. На данном дисплее это заметно, он не является особо качественным. Для этого предусмотрен пункт «сон», который затемняет дисплей. Это продлит срок службы дисплея.

1. Планируется добавить возможность настройки времени в меню Serial по протоколу RS-232 (последовательный порт), или же в режиме ввода данных напрямую через энкодер.
2. Планируется добавить взаимодействие с EEPROM памятью, чтобы сохранять значения настроек.
3. Планируется разработать печатную плату и заказывать ее у промышленных изготовителей, чтобы упросить процесс сборки.

**Сборка**

Собираю

**Экономическая составляющая:**

Энкодер: 70 рублей

Модуль реального времени: 85 рублей

Элемент питания CR2032: 50 рублей

Arduino: 200 рублей

OLED: 150 рублей

Реле: 100 рублей

Тумблер: 30 рублей

Прочие расходы: 100 рублей

Итог: 785 рублей

К итоговой стоимости можно добавить цену печатной платы: при заказе двадцати штук стоимость не превысит 150 рублей.

Похожие системы с менее наглядной индикацией стоят больше пяти тысяч рублей.

Плюсом моей системы является возможность доработки: все ошибки в работе системы можно исправить, имея компьютер. А при желании и вовсе полностью переписать код, сделать по-своему.

Итог:

Я считаю, что мой проект действительно сможет помочь моему лицею и другим школам. В случае использования подобных систем освободится человеческий ресурс, который можно будет перенаправить на более важные и сложные задачи. Я приобрел новый опыт в использовании микроконтроллеров, научился взаимодействовать с дисплеями