ОПИСАНИЕ И НАСТРОЙКА

Паяльной станции BGA от Watashi

Краткое описание элементов:

Платформа - Apduino Mega2560, дисплей TFT с тачем или без(совместимый с Мегой), клавиатура резистивная аналоговая или обычные кнопки, энкодер (лучше использовать с готовой обвязкой), модуль PTC часов DS3231 или DS1307, детектор нуля (кто решит его поставить), реле к1-к4 по для подключения - отключения сегментов нижнего подогрева, два модуля преобразователя термопары К-типа на max6675, верхний и нижний нагреватели, два модуля твердотельных реле типа SSR и бузер-пищалка.

Arduino IDE

Наверно не надо объяснять, зачем нужно использовать последнюю версию IDE? Лучше скачать и установить с сайта производителя <https://www.arduino.cc/en/Main/Software>

Где скачать скетч?

Можно тут: <https://clck.ru/JR9NJ>

или тут: <https://drive.google.com/open?id=1zKl0bH17UEs_OnylpdnUQM2NhLSQ51JG>

(лучше скопировать ссылку на гугл-диск в браузер и открыть)

Порядок настройки

Настроить клавиатуру инструкция ниже, тоже тачпад и энкодер.

Настроить время.

Ввести исходные данные:

* Основной файл прошивки Svs\_V08.ino (в папке с ним должены находится все прилагаемые файлы).
* В файле Setting.h произвести настройки на нужные пины и включить-выключить необходимые функции.
* Исправить в файле ProfDate.c исходные данные профилей, загружаемые в EEPROM.
* Проверить и откорректировать (если требуется) настройки профилей.
* Приступать к испытаниям )))

Дисплей

* Используется библиотека UTFT с добавленным русским языком.
* Скачать вариант от Dmitrish тут [ссылка](http://forum.amperka.ru/threads/%D0%98%D0%9A-%D0%BF%D0%B0%D1%8F%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F-%D1%81%D1%82%D0%B0%D0%BD%D1%86%D0%B8%D1%8F-%D0%BD%D0%B0-arduino-mega-2560-%D0%94%D0%BE%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0-%D1%81%D0%BA%D0%B5%D1%82%D1%87%D0%B0-ars_v2_lilium_jsn.10176/page-162#post-243225)
* или мой вариант с исправлениями и добавлениями тут: <https://drive.google.com/open?id=1EbPZPjbRe86g-V2yyKmuwKZ21ZdFYWkX>

Я использую дисплей такой: [ссылка](https://ru.aliexpress.com/item/32833726210.html?spm=a2g0s.9042311.0.0.274233edPKd2An) (похоже ссылка уже не актуальна).

Есть еще подходящий дисплей на 3.95” про его приобретение и использование [тут](http://forum.amperka.ru/threads/%D0%98%D0%9A-%D0%BF%D0%B0%D1%8F%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F-%D1%81%D1%82%D0%B0%D0%BD%D1%86%D0%B8%D1%8F-%D0%BD%D0%B0-arduino-mega-2560-%D0%94%D0%BE%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0-%D1%81%D0%BA%D0%B5%D1%82%D1%87%D0%B0-ars_v2_lilium_jsn.10176/page-163#post-243327).

https://aliexpress.ru/item/4000126450233.html

В скетче он обьявляется такой строкой: UTFT myGLCD(CTE40, 38, 39, 40, 41);

Отлаживайте сначала на стандартных примерах из библиотеки, а потом строчку объявления дисплея переносите в скетч.

Клавиатура

В файле Setting.h раскомментировать строчку в соответствии с используемой клавиатурой

Код (C++):

#define SetAnalogBatton // Если используется аналоговая клавиатура

//#define SetDigitBatton // Если используется цифровая клавиатура

Цифровые кнопки подключаются к выводам А11-А15 одним концом (к нему же подключается подтягивающий резистор 10К к питанию), другой на земле.

Аналоговая клавиатура. Перед началом работы ее нужно настроить вот эти константы в файле Setting.h

// ожидаемые значения для псевдо-кнопок  
#define SetRIGHT 144  
#define SetUP 353  
#define SetDOWN 711  
#define SetLEFT 569  
#define SetSELECT 4

Настраиваются следующим образом:

* находим в основном скетче

Код (C++):

void loop() {  
  unsigned long currentMillis = millis();  
  // Serial.println(analogRead(Knopki));

* раскомментируем Serial ...,
* загружаем скетч,
* включаем Монитор порта,
* нажимаем клавиши и получаем числа,
* заполняем соответственно #define (см выше), значения могут немного плавать, но это не беда, допускается отклонение плюс-минус 10, значения должны быть в диапазоне 0-1023,
* закомментируем Serial ...,
* загружаем скетч,
* пробуем ))

Энкодер

Мой небольшой опыт работы с энкодером показал, что часто попадаются глючные, при вращении ручки иногда срабатывают со второго и даже третьего щелчка.

Чтобы это исправить, воспользовался опытом AlexGyver и скрестил его библиотеку с RotaryEncoder-master и получилась RotaryEncoder-Gyver. В файле Encoder.h есть настройки:

// Введены два новых параметра, которые могут принимать два значения 0 или 1

#define Rotate 1  // направление вращения

#define I\_Type 0  // тип энкодера

// если энкодер ведет себя неправильно - смените тип энкодера

//

RotaryEncoder encoder(Enc\_Clk, Enc\_DT, Rotate, I\_Type);   // пины подключения энкодера

На стандартной китайской платке не запаян один резистор 10ком, я его запаял. Подправил библиотеку Cl\_do\_btn\_long для использования внутреннего подтягивающего резистора (чтобы не паять), хотя сам пока не проверил. Проверил – не работает, нужен подтягивающий резистор.

Touch-Screen (у кого есть )

Я столкнулся с двумя видами тача на дисплеях:  
1 - используется два цифровых выхода с ардуино и два аналоговых. Используется библиотека Touch-Screen-Library-master, предназначена для Uno или Mega.  
2 - используется интерфейс типа SPI и использует библиотеку URTouch.

В начале файла Touch.h есть две строчки:

Код (C++):

#define SetTouch1 // влючается тач где используются аналоговые сигналы

// #define SetTouch2 // включается тач где используется интерфейс SPI

Раскомментирование одной из них включает соответствующий тип тача. Если оба закомментированы тач не используется.

Исходные данные ( профили )

* В моей программе используется структура исходных данных отличная от скетча Dmitrysh, поэтому:  
  что бы данные не пересекались и программы не мешали друг другу, начальный адрес записи в EEPROM равен 1000. Его можно изменить строчкой в скетче

#define Adres 1000 // адрес начала записи в EPROM.

* С программой идет файл ProfDate.c, можно поменять и заголовок профиля (только русские буквы и цифры, длина ограничена строкой на дисплее -2 символа) и данные в самом профиле. (**только аккуратно с запятыми, не убирать и не добавлять - это существенно**). Можно использовать программу Notepad++ (кодировка UTF-8) или сразу в Arduino IDE и не думать про кодировки.
* После компиляции и запуска основной программы, зайдя в «настройки» – вкладка «ОБЩИЕ» - «загрузка профилей», можно загрузить предустановленные профили из файл ProfDate.c в EEPROM и можно переходить дальнейшей работе.
* если кому то мало 10 профилей (или слишком много) можно изменить их количество в строчке:

#define numMax 10 // Максимальное количество профилей

Часы, настройка

Изначально думал что будет логирование данных с записью на SD карту и часы пригодились бы несомненно, а получилось пока только украшалка )).

Библиотеку подбирал чтобы работали два модуля DS1307 и DS3231

Второй модуль предпочтительней как более новый и более точный в работе.

Установка часов. Производится в настройках во вкладке «ОБЩИЕ».

Управление реле.

В файле Setting.h есть строчка влияющая на логику работы реле:

#define SetReleNull // если раскомменчена то реле включаются "0", иначе "1"

Корректировка значений термопар

Если вы произвели калибровку используемых термопар и у вас есть расхождения с эталонными значениями, особенно в диапазоне 200 – 230 градусов, т.е. температуры плавления припоя, можно откорректировать, введя разницу значений в следующие строчки:

Код (C++):  
// вместо 0 ставим +- число

#define Corect\_T 0 // термопара верхнего нагревателя

#define Corect\_B 0 // термопара нижнего нагревателя

Иконки на кнопках на главном экране

Код (C++):  
// #define Set\_Picture  // Если нужно выводить иконки

раскоментирйте строку с Set\_Picture, при этом файл Graf.с должен быть в одной папке со скетчем и после компиляции и заливке скетча они появятся.

Основная программа

Старался делать интерфейс интуитивно понятным, большинство действий клавиатурой, энкодером и тачем показано на видео на канале в [**YouTube**](https://www.youtube.com/user/watashisolify/videos)**.**

С моей стороны есть попытка сделать универсальный алгоритм управления станцией под любые нагреватели. Отсюда и большое количество разных настроек. Было несколько вариантов, но предложенный алгоритм у меня показал лучшие результаты. Хотя меня он не удовлетворяет.  
Алгоритм управления станцией следующий:

* Рассчитывается время с текущей температуры до температуры удержания низа с учетом скорости нагрева низа. Включается низ и ПИДу выдается температура, которую он должен выдержать, согласно графика роста температуры. По окончании времени роста температуры, Пиду выдается температура удержания низа и это отрабатывается до конца пайки. Если реальная температура сильно отличается от задаваемой (падает или не растет), то это признак аварийной ситуации и выключения процесса пайки. Последнее еще только в планах.
* Преднагрев верха включается по достижении показаний датчика верха температуры удержания низа. Время преднагрева задается параметром "ВРЕМЯ ЗАДЕРЖКИ" настроек низа. Мощность преднагрева устанавливается параметром "МОЩНОСТЬ МИН" в 1 шаге настроек верха.
* По окончании времени работы преднагрева, запускается верх. Первый шаг: Включается верх и ПИДу выдается температура, которую он должен выдержать, согласно графика, роста температуры, затем полочка задается параметром "ВРЕМЯ ПЕРЕХОДА", где удерживается температура, заданная в текущем шаге.
* По окончании полочки идет переход на следующий шаг. После отработки последнего шага остановка процесса пайки.
* Окончание профиля задается "0" в параметре "Скорость нагрева" в настройках шагов верха.

Хотелось бы добавить следующее, что этот алгоритм - это компактный кусок кода в одном месте. Можно спокойно вставить какой-то другой алгоритм и поставить переключатель этих алгоритмов (хоть десяток, ограничения только в самой MEGA).

Связь с компьютером

Если в файле Setting.h раскомментировать строчку:

Код (C++):  
//#define SetConnectPC // Закоментировать, если вообще не надо связь с РС

То скетч будет посылать данные о текущей температуре. Для этого надо запустить на компьютере программу irsp.ехе, ввести используемый COM-порт, соединиться и получим вывод текущей температуры в виде графика.

Меню настроек

«РАЗМЕР НИЗА» - без размерности, диапазон 1-3,

«СКОРОСТЬ НАГРЕВА» - градусы в секунду, диапазон от 0 до 3, через 0.1

Нулевое значение означает конец профиля, если ноль в первом шаге для ВЕРХА, то будет работать только НИЗ, и будет удерживать установленную температуру. Надо добавить, что в памяти EEPROM этот параметр хранится как целое число и занимает 1 байт, а отображается как число с запятой. Редактируется как число без запятой, но в последующих расчетах запятая учитывается.

«ТЕМПЕРАТУРА» - градусы, диапазон 0-255,

«ВРЕМЯ ЗАДЕРЖКИ» - в секундах

«ВРЕМЯ ПЕРЕХОДА» - в секундах, это время задержки после роста температуры в текущем шаге до начала роста в следующем шаге. Температура в это время удерживается постоянная.

«МОЩНОСТЬ МИН» в процентах, диапазон 0-99 для соответствующего нагревателя.

«МОЩНОСТЬ МАХ» в процентах, диапазон 0-99 для соответствующего нагревателя.

Коэффициенты ПИД регулятора «ВЕРХ ПРОПОРЦИОНАЛ» «ИНТЕГРАЛ» «ДИФФЕРЕНЦИАЛ» «НИЗ\_ПРОПОРЦИОНАЛ» - без размера, диапазон 0-255.