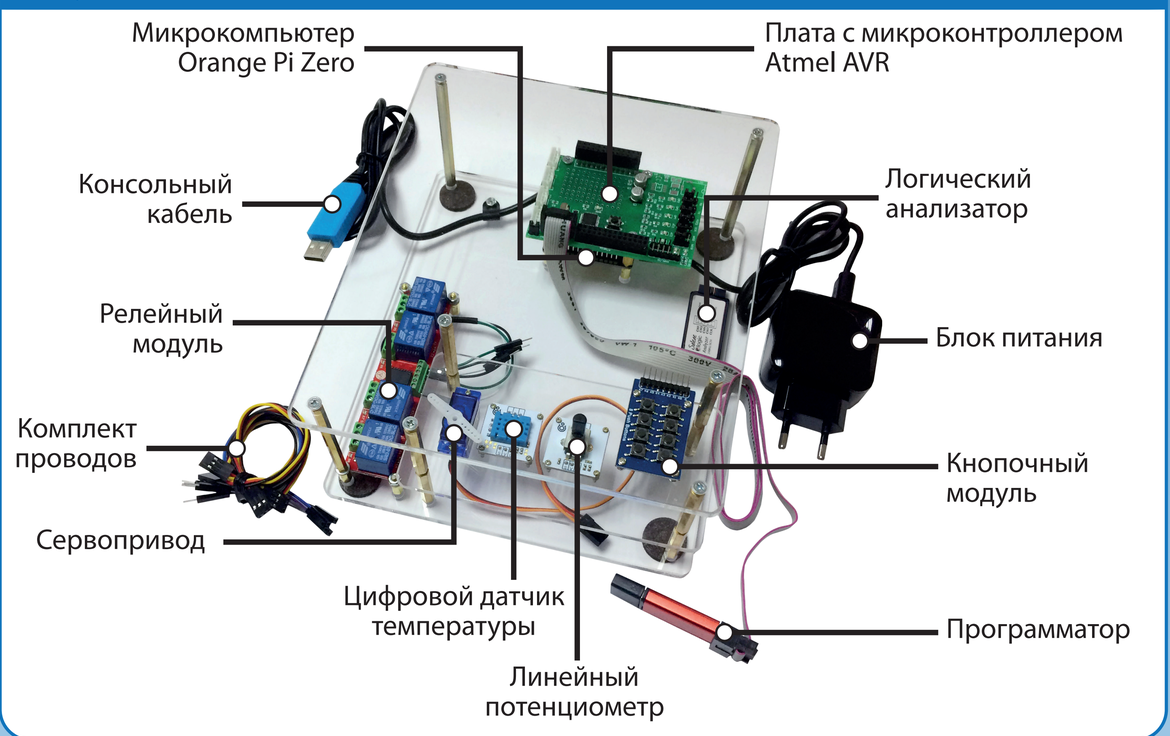
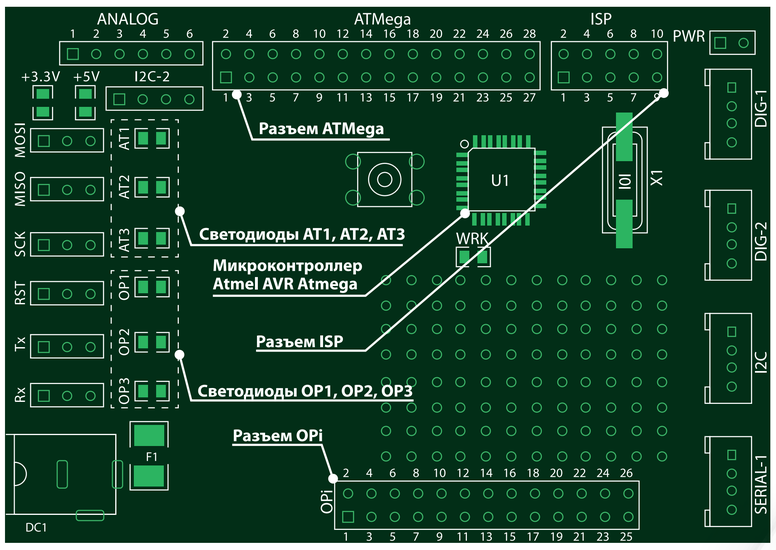
D-Link® Учебный стенд DTK-1

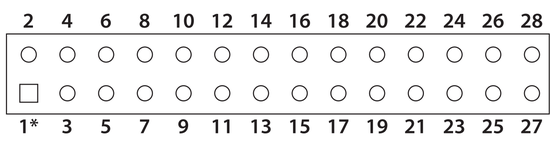
Вид учебного стенда и его состав



Плата расширения с микроконтроллером Atmel AVR и ее разъемы ATMega и OPi



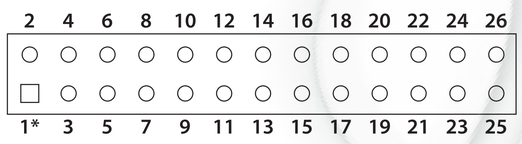
Выводы разъема ATMega



№	Порт	Arduino	Функция
1			+5 В
2	PD1	1	TxD
3			GND
4	PD0	0	RxD
5			Не используется
6			Reset
7			+3,3 В
8			GND
9	PC3	A3	
10	PD2	2	INT0
11	PC2	A2	
12	PD3	3	INT1 / PWM
13	PC1	A1	
14	PD4	4	T0 / XCK
15	PC0	A0	
16	PD5	5	T1 / PWM
17	PB5	13	SCK
18	PD6	6	AIN0 / PWM
19	PB4	12	MISO
20	PD7	7	AIN1
21	PB3	11	MOSI / OC2 / PWM
22	PB0	8	ICP
23	PB2	10	OC1B / SS / PWM
24	PB1	9	OC1A / PWM
25			ADC7
26			ADC6
27	PC5	A5	SCL
28	PC4	A4	SDA

1* первый вывод на схемах часто обозначается квадратом
№ номер вывода в соответствии со схемой
Порт обозначение вывода порта микроконтроллера AVR
Arduino обозначение вывода в среде Arduino IDE (язык Wiring)
Функция функция (назначение) вывода микроконтроллера

Выводы разъема OPi



№	Порт	Wiring	Функция
1			3.3V – Out in board
2			5V - Input
3	PA12	8	TWI0_SDA / GPIO12
4			5V - Input
5	PA11	9	TWI0_SCK / GPIO11
6			GND
7	PA06	7	PWM1 / GPIO6
8	PG06	15	UART1_TX / GPIO198
9			GND
10	PG07	16	UART1_RX / GPIO199
11	PA01	0	UART2_RX / GPIO1
12	PA07	1	SIM_CLK / PA_EINT7 / GPIO7
13	PA00	2	UART2_TX / GPIO0
14			GND
15	PA03	3	UART2_CTS / GPIO3
16	PA19	4	TWI1_SDA / GPIO19
17			3.3V – Out in board
18	PA18	5	TWI1_SCK / GPIO18
19	PA15	12	SPI1_MOSI / GPIO15
20			GND
21	PA16	13	SPI1_MOSI / GPIO16
22	PA02	6	UART2_RTS / GPIO2
23	PA14	14	SPI1_MOSI / GPIO14
24	PA13	10	SPI_CS / GPI013
25			GND
26	PA10	11	SIM_DET / PA_EINT10 / GPIO10

1* первый вывод на схемах часто обозначается квадратом
№ номер вывода в соответствии со схемой
Порт обозначение вывода порта процессора Allwinner H2+/H3
Wiring обозначение вывода в библиотеках WiringPi
Функция функция (назначение) вывода процессора Allwinner H2+/H3

Функции библиотеки Arduino Wiring (для МК AVR)

Основные функции

pinMode (num, mode) — Выбор режима работы вывода с номером **num** (параметр **mode** может принимать значения **INPUT** – ввод, **INPUT_PULLUP** – ввод с подтяжкой вверх, **OUTPUT** – вывод)

digitalWrite (int num, int value) — Запись значения **value** (принимает значение **HIGH** (**1**) или **LOW** (**0**) на вывод с номером **num**

int value = digitalRead (int num) — Передает значение (**HIGH** или **LOW**) с вывода **num** в переменную **value**

void delay (int ms) — Останавливает работу контроллера на время **ms** (миллисекунды)

Функции для работы с аналоговыми интерфейсами

int value=analogRead (pin) — Считывает значение с указанного аналогового входа pin в переменную value

analogReference (type) — Определяет опорное напряжение, относительно которого происходят аналоговые измерения. Значения переменной type: **DEFAULT**, **INTERNAL**, **INTERNAL1V1**, **INTERNAL2V56**, **EXTERNAL**

analogWrite (pin, value) — Устанавливает аналоговое значения на выводе. Используется при формировании широтно-импульсной модуляции

Функции для работы с интерфейсом UART

Serial.begin (speed) — Инициализация последовательного порта, значения параметра **speed** (определяет скорость передачи данных): **300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600** или **115200**

Serial.end () — Закрытие последовательного порта

Serial.available () — Возвращает количество байт, полученных контроллером **UART** (данные хранятся в буфере FIFO)

Serial.read () — Чтение байта из буфера порта. Если буфер пуст, то возвращается значение **-1**

Serial.flush () — Ожидание окончания передачи исходящих данных

Serial.print (val, format) — Передает данные через последовательный порт как ASCII текст. Параметры функции: **val** - данные для передачи; **format** (не обязательный) – базис для целых чисел или количество знаков после запятой для вещественных

Serial.println (val, format) — Передает данные через последовательное соединение как ASCII текст с следующим за ним символом переноса строки (ASCII символ **13** или **'\r'**) и символом новой строки (ASCII **10** или **'\n'**)

Serial.write () — Функция передает данные в виде бинарного кода через последовательное соединение. Данные посылаются как один или серия байтов

Serial.peek () — Возвращает следующий доступный байт (символ) из буфера входящего последовательного соединения, не удаляя его из этого буфера

Функции для работы с интерфейсом I²C

Wire.begin (int address) — Инициализирует библиотеку **Wire** и подключается к шине I²C как ведущий (мастер) или ведомый. Параметр **address** – 7-битный адрес ведомого устройства (необязательный), если не задан, плата подключается к шине как мастер

Wire.beginTransmission () (int address) — Открывает канал связи по шине I²C с ведомым устройством. Параметр **address** – 7-ми битный адрес устройства к которому посылается запрос (обязательный параметр)

Wire.endTransmission () (bool stop) — Отправляет данные, которые были поставлены в очередь методом **write ()** и завершает передачу. Параметр **stop** – может принимать 2 значения **true** – после запроса отправляет **STOP**, освобождая шину I²C, **false** – после запроса отправляет **RESTART**, при этом шина не освобождается и можно отправлять дополнительные запросы

Wire.write (int data) — Ставит данные в очередь для передачи. Параметры: **data** - байт, срока или массив байтов для передачи; **length** –длина передаваемого массива данных

int Wire.read () — Считывает байт принятой контроллером I²C информации

int Wire.available () — Возвращает количество байт, доступных для чтения

byte Wire.requestFrom (address, quantity, stop) — Отправляет запрос на определенное количество байт от ведущего устройства к ведомому. Параметры: **address** – 7-ми битный адрес устройства к которому посылается запрос (обязательный); **quantity** – количество запрашиваемых байт ; **stop** – тип данных boolean, **true** – после запроса отправляет **STOP**, освобождая шину I²C, **false** – после запроса отправляет **RESTART**, при этом шина не освобождается и можно отправлять дополнительные запросы, по умолчанию – **true**. Возвращаемые значения **byte** (указывает на состояние передачи): **0** - успешная передача; **1** - объем данных для передачи слишком велик; **2** - принят NACK при передаче адреса; **3** - принят NACK при передаче данных; **4** - другие ошибки

Wire.setClock (clockFrequency) — Определяет тактовую частоту обмена данными. Параметр **clockFrequency** – новое значение частоты обмена данными в герцах. Доступные значения: **10000, 100000, 400000, 1000000, 3400000**

Wire.onReceive (procedure) — Определяет функцию обработчик прерывания, которая будет выполняться при получении данных от ведущего устройства

Wire.onRequest (procedure) — Определяет функцию обработчик прерывания, которая будет выполняться при получении запроса от ведущего устройства

Функции библиотеки WiringOP (для Orange Pi Zero)

Функции инициализации

wiringPiSetup (void) — Инициализирует библиотеку WiringPi (использует упрощенную схему контактов WiringPi, которая привязывает GPIO-контакты к виртуальным номерам от «0» до «16»)

wiringPiSetupGpio (void) — Инициализирует библиотеку WiringPi (использует GPIO-контакты напрямую, без привязки к виртуальным контактам)

wiringPiSetupPhys (void) — Инициализирует библиотеку WiringPi (использует физические номера контактов)

wiringPiSetupSys (void) — Инициализирует библиотеку WiringPi (обеспечивает прямой доступ к компонентам Pi, использует интерфейс **/sys/class/gpio**)

Основные функции

void pinMode (int pin, int mode) — Задает режим работы контакта. Допустимые варианты – **INPUT**, **OUTPUT**, **PWM_OUTPUT** и **GPIO_CLOCK**.

void pullUpDnControl (int pin, int pud) — Включает у контакта, который должен быть настроен на режим **INPUT**, подтягивающий вверх или вниз резистор. Вместо параметра **pud** нужно поставить либо **PUD_OFF** (все резисторы отключены), либо **PUD_DOWN** (резистор притягивает сигнал к «земле»), либо **PUD_UP** (резистор подтягивает сигнал к 3,3 вольтам)

void digitalWrite (int pin, int value) — Записывает на указанный контакт, который должен быть настроен на режим **OUTPUT**, значение **HIGH** или **LOW** (т.е. «**1**» или «**0**»)

int value = digitalRead (int pin) — Возвращает значение, считанное с указанного контакта. Это может быть **HIGH** или **LOW** («**1**» или «**0**») – в зависимости от логического уровня, который в данный момент имеется на контакте

void digitalWriteByte (int value) — Записывает в первые восемь GPIO-контактов указанный байт

Функции для работы с интерфейсом UART

int fd = serialOpen (device, int baud) — Инициализация последовательного порта, параметр **baud** определяет скорость передачи данных и может принимать значения: **300**, **1200**, **2400**, **4800**, **9600**, **14400**, **19200**, **28800**, **38400**, **57600** или **115200**. Параметр **device** – имя устройства (путь к «файлу» устройства). Функция возвращает указатель открытого порта или **-1** при возникновении ошибки

serialClose (fd) — Закрывает последовательный порт с указателем **fd**

serialPutchar (fd, databyte) — Отправляет байт данных через порт с указателем **fd**

void serialPuts (fd, databytes) — Отправляет набор байт данных через порт с указателем **fd**

void serialPrintf (fd, message, …) — Эмулирует функцию **printf** при работе с UART

Работа с микроконтроллером AVR

Настройка оборудования в среде Arduino IDE

- Инструменты > Плата > Arduino Pro or Pro Mini
- Инструменты > Процессор > ATmega328P (3.3V, 8MHz)
- Инструменты > Программатор > USBASP

Сборка и загрузка программы

- Скетч > Проверить/Компилировать
- Скетч > Загрузка через программатор (Ctrl+Shift+U)

Работа с микрокомпьютером Orange Pi Zero

Подключение с помощью консольного кабеля (Linux)

sudo minicom -c on -b 115200 -D /dev/ttyUSB0 или
sudo picocom --imap lfcrlf -b 115200 /dev/ttyUSB0

Компиляция программы на языке C

gcc имя_программы.c -o имя_программы.out -lwiringPi -lpthread

Запуск программы на исполнение

sudo ./имя_программы.out

Установка Библиотеки Wiring

git clone https://github.com/xpertsavenue/WiringOP-Zero.git
cd WiringOP-Zero
chmod +x ./build
sudo ./build

Настройка IP-адреса микрокомпьютера

sudo ip link set eth0 up
sudo ip addr add 192.168.1.1/24 dev eth0

Copyrigth © 2025 ООО "Д-Линк Трейд". Все права защищены. www.dlink.ru. Адрес: РФ, Рязанская область, г. Рязань, проезд Шабулина 16. Тел./Факс: +7 (495) 744 00 99. Портал дистанционного обучения и сертификации D-Link: http://learn.dlink.ru/login/.