Arduino кітапханасы дегеніміз не?

• Arduino ортасы көптеген бағдарламалау платформалары сияқты кітапханаларды пайдалану арқылы кеңейтілуі мүмкін.

• Сіз өзіңіздің жеке кітапханаларыңызды жасай аласыз

• Бұл сіздің кодты құрудың жақсы тәсілі

• Әр түрлі қосымшалар мен жобаларда кодты бөлісудің және қайта қолданудың жақсы әдісі

• Жеке кітапханаңызды құру оңай

• Негізінен Arduino кітапханасы - бұл бір немесе бірнеше функцияны қамтитын C ++ сыныбы

• PID контроллерін, масштабтауды және Lowpass сүзгісін Arduino кітапханасына, яғни 3 функциясы бар класта орналастыру ұсынылады.

Стандартные библиотеки Ардуино

Начать знакомство с библиотеками лучше с официального сайта, на котором можно найти внушительный список стандартных модулей и ссылки на официальные библиотеки партнеров.

Список встроенных библиотек (они поставляются вместе с дистрибутивом Arduino IDE):

* EEPROM
* Ethernet / Ethernet 2
* Firmata
* GSM
* LiquidCrystal
* SD
* Servo
* SPI
* SoftwareSerial
* Stepper
* TFT
* WiFi
* Wire

[Официальная страница на сайте Arduino](https://www.arduino.cc/en/Reference/Libraries)

Подборка библиотек в одном архиве

Если у вас нет времени на детальный анализ множества сайтов и вы хотите скачать все необходимое для работы с внешними устройствами Ардуино в одном архиве, мы подготовили список 40 самых популярных библиотек. Просто [скачайте архив](https://arduinomaster.ru/wp-content/uploads/2018/02/libraries.zip) и распакуйте его содержимое (папку libraries) в папку Arduino.

Библиотеки для экранов, индикаторов и дисплеев

Библиотека I2C

Скачать: <https://github.com/g4lvanix/I2C-master-lib>

Библиотека, предназначенная для работы периферийного устройства по протоколу I2C.

**Пример использования:**

#ifndef I2C\_MASTER\_H

#define I2C\_MASTER\_H

void I2C\_init (void) – создание объекта, настройка на правильную частоту для шины.

uint8\_t I2C\_start () – установка соединения с новым устройством.

uint8\_t I2C\_write() – запись данных на текущее устройство.

uint8\_t I2C\_read\_ack() – считывание байта с устройства, запрос следующего байта.

Библиотека  LiquidCrystal

Скачать: <https://github.com/mk90/LiquidCrystalRus>

Стандартная библиотека, установленная в Arduino IDE. Предназначена для управления жидкокристаллическими дисплеями LCD.

**Пример использования:**

#include <LiquidCrystal.h>. Также, чтобы не ошибиться при написании, можно подключить через меню Sketch – Import Library – LiquidCrystal.

Конструктор класса –  LiquidCristal(…). Аргументами являются rs, rw, en, do…d7. Первые 3 соответствую выводам сигналов RS, RW и Enable. Выводы d соответствуют номерам шин данных, к которым подключен дисплей.

void begin(cols, rows) – метод, который инициализирует интерфейс дисплея. Аргументами являются количество знаков в строке (cols) и число строк (rows). Этот метод должен задаваться первым.

void createChar(num, data) – метод, необходимый для создания пользовательских символов.

Библиотека UTFT

Скачать: <https://github.com/amperka/UTFT>

Стандартная библиотека, необходимая для работы Ардуино с TFT экранами разных типов. Все поддерживаемые дисплеи представлены в сопроводительном документе с библиотекой.

**Пример использования:**

#include <UTFT.h>

UTFT(); – создание экземпляра UTFT.

textRus(char\*st, int x, int y); – метод, позволяющий выводить строку из указателя. Например, char \*dht = “Температура,С”;

textRus(string st, int x, int y); – вывод строки с указанием в параметре. Например, g.textRus(“Температура, С”, 0, 20);

Библиотека LedControl

Скачать: <https://github.com/wayoda/LedControl>

Позволяет управлять семисегментными дисплеями, объединять массив из светодиодов в одну матрицу.

**Пример использования:**

#include <LedControl.h>

LedControl  lc1= LedControl( );

– требуется для инициализации библиотеки. Должна состоять из четырех аргументов – номера пинов, к которым подключен дисплей (первые 3 аргумента) и количество подключенных чипов.

writeArduinoOn7Segment() – отображение на дисплее всех чисел от 0 до 15. Использует функции setChar() для символов a и d и setRow() для создания макета пропущенных символов.

LedControl.shutdown() – отключение изображения.

setIntensity() – контроль яркости.

Библиотеки для работы с датой и временем ардуино

Библиотека RTClib

Скачать: <https://github.com/adafruit/RTClib>

Библиотека для работы с часами реального времени, упрощающая взаимодействие с Ардуино.

**Пример использования:**

#include <RTClib.h>

RTC\_DS1307 RTC; – выбор датчика (в данном случае DS1307).

rtc.adjust(DateTime( Date, Time)); – настройка времени и календаря.

dayOfTheWeek () – вывод дня недели. Аргумент от 0 до 6, 0 – воскресенье.

Библиотека Timelib

Скачать: <https://github.com/PaulStoffregen/Time>

Позволяет Ардуино получать информацию о дате и времени в данный момент.

**Пример использования:**

#include <TimeLib.h>

Time(); – создание экземпляра.

setTime (t); – установка времени. Аргумент t – час, минута, секунда, день, месяц и год.

timeStatus(); – показывает, установлено ли время.

adjustTime(adjustment); – настройка времени.

Библиотека Ds1307

Скачать: <https://github.com/PaulStoffregen/DS1307RTC>

Библиотека для удобного взаимодействия часов DS1307 с Ардуино c использованием библиотеки Wire.

**Пример использования:**

#include <DS1307RTC.h>

class DS1307RTC – создание объекта DS1307.

SetTime() – установка времени.

get() – считывает RTC, возвращает полученную дату в формате POSIX.

Set(time\_t t) – запись даты в RTC

Библиотека DS 3231

Скачать: <https://github.com/rodan/ds3231>

Предназначена для управления датой и временем в модуле ds3231.

#include “ds3231.h”

DS3231  Clock(SDA, SCL); – создание объекта DS3231, подключение к линии тактирования и линии данных.

getTime(); – считывание даты и времени с часов.

setDate(date, mon, year); – установка даты.

Системные библиотеки ардуино

Библиотека EEPROM

Скачать: <https://github.com/aterentiev/EEPROM2>

Стандартная библиотека. Предназначена для работы с энергонезависимой памятью (запись данных, их чтение).

**Пример использования:**

#include <EEPROM.h>

EEPROM.read(); – создание объекта, считывание байта по адресу из энергонезависимой памяти.

EEPROM.write(address, value)– запись байта в энергонезависимую память.

EEPROM.put() – запись строк чисел с плавающей запятой.

EEPROM.get() – чтение строк и чисел с плавающей запятой.

Библиотека SoftwareSerial

Скачать: <https://github.com/PaulStoffregen/SoftwareSerial>

Библиотека, которая позволяет реализовывать последовательные интерфейсы с любых цифровых пинов. Также позволяет создавать несколько последовательных портов, которые работают на скорости до 115200 бод.

#include <SoftwareSerial.h>

SoftwareSerial mySerial(RX, TX) – создание объекта, аргументы – выводы, к которым подключены RX и TX.

Serial.begin( ); – устанавливает скорость порта для связи ардуино и компьютера.

mySerial.overflow() – проверка входного буфера на переполнение.

Библиотека Math

Скачать: <http://www.nongnu.org/avr-libc/user-manual/group__avr__math.html>

Включает в себя большое количество математических функций для работы с числами с плавающей запятой.

**Пример использования:**

#include <math.h>

Math(); – создание экземпляра Math.

Serial.print(“cos num = “); – возвращает косинус числа.

Serial.println (fmod (double\_\_x, double\_\_y)); – возвращает числа по модулю.

Библиотека Scheduler

Скачать: <https://www.arduino.cc/en/Reference/Scheduler>

Предназначена для работы с Arduino Due, позволяет работать в режиме многозадачности. Пока является экспериментальной библиотекой.

**Пример использования:**

#include <Scheduler.h>

Scheduler; – создание экземпляра.

Scheduler.startLoop() – позволяет добавить функцию, которая будет выполняться вместе с loop().

yield() – позволяет передать управление другим задачам.

Библиотеки серво моторов и шаговых двигателей

Библиотека Servo

Скачать: <https://github.com/netlabtoolkit/VarSpeedServo>

Стандартная библиотека. Необходима для управления серводвигателями  и часто используется в робототехнических проектах с манипуляторами.

**Пример использования:**

#include <Servo.h>

Servo myservo; – создание объекта для серводвигателя..

myservo.attach(); – номер выхода, к которому подключен серводвигатель.

myservo.write(180, 30, true); – движение на 180 градусов, скорость 30, ожидание окончания движения.

Библиотека Stepper

Скачать: <https://www.arduino.cc/en/Reference/Stepper?from=Tutorial.Stepper>

Небходима для управления шаговым униполярным и биполярным двигателем.

#include <Stepper.h>

const int stepsPerRevolution =  ; – количество шагов, за которое двигатель проходит полный поворот.

Stepper myStepper = Stepper(steps, pin1, pin2) – создает экземпляр класса с указанным количеством шагов и выводами, к которым подключается двигатель.

Библиотеки датчиков ардуино

Библиотека DHT

Скачать: <https://github.com/adafruit/DHT-sensor-library>

Библиотека, которая позволяет считать данные с температурных датчиков DHT-11 и DHT-22.

#include < DHT.h>

DHT dht(DHTPIN, DHT11); – инициализирует датчик (в данном случае DHT11).

dht.begin(); – запуск датчика.

float t = dht.readTemperature(); – считывание текущего значения температуры в градусах Цельсия.

Библиотека DallasTemperature

Скачать: <https://github.com/milesburton/Arduino-Temperature-Control-Library>

Предназначается для работы с датчиками Dallas. Работает совместно с библиотекой OneWire.

#include <DallasTemperature.h>

DallasTemperature dallasSensors(&oneWire); – передача объекта oneWire для работы с датчиком.

requestTemperatures() – команда считать температуру с датчика и

положить ее в регистр.

printTemperature(sensorAddress); – запрос получить измеренное значение температуры.

Библиотека Ultrasonic

Скачать: <https://github.com/JRodrigoTech/Ultrasonic-HC-SR04>

Обеспечивает работу Ардуино с ультразвуковым датчиком измерения расстояния HC-SR04.

#include <Ultrasonic.h>

Ultrasonic ultrasonic (tig , echo) – объявление объекта, аргументы – контакт Trig и контакт Echo.

dist = ultrasonic.distanceRead(); – определение расстояния до объекта. Агрумент – сантиметры(СМ) или дюймы (INC).

Timing() – считывание длительности импульса на выходе Echo, перевод в необходимую систему счисления.

Библиотека ADXL345

Скачать: <https://github.com/adafruit/Adafruit_ADXL345>

Предназначается для работы с акселерометром ADXL345.

**Пример использования:**

#include <Adafruit\_ ADXL345.h>

ADXL345\_ADDRESS – создание объекта, указание его адреса.

ADXL345\_REG\_DEVID  – идентификация устройства.

ADXL345\_REG\_OFSX – смещение по оси Х.

ADXL345\_REG\_BW\_RATE – управление скоростью передачи данных.

Библиотека BME280

Скачать: <https://github.com/adafruit/Adafruit_BME280_Library>

Предназначается для работы с датчиком температуры, влажности и давления BME280.

**Пример использования:**

#include <Adafruit\_BME280.h>

BME280\_ADDRESS  – создание объекта BME280, указание его адреса.

begin(uint8\_t addr = BME280\_ADDRESS); – начало работы датчика.

getTemperature – получение измеренной температуры.

getPressure – получение измеренного давления.

Библиотека BMP280

Скачать: <https://github.com/adafruit/Adafruit_BMP280_Library>

Требуется для работы с датчиком атмосферного давления BMP280.

**Пример использования:**

#include <Adafruit\_BMP280.h>

BMP280\_CHIPID – создание экземпляра, указание его адреса.

getTemperature(float \*temp); – получение измеренной температуры.

getPressure(float \*pressure); – получение измеренного значения давления.

Библиотека BMP085

Скачать: <https://github.com/adafruit/Adafruit-BMP085-Library>

Требуется для работы с датчиком давления BMP085.

**Пример использования:**

#include <Adafruit\_BMP085.h>

Adafruit\_BMP085 bmp; – создание экземпляра BMP085.

dps.init(MODE\_ULTRA\_HIGHRES, 25000, true); – измерение давления, аргумент 25000 – высота над уровнем моря (в данном случае 250 м. над уровнем моря).

dps.getPressure(&Pressure); – определение давления.

Библиотека FingerPrint

Скачать: <http://iarduino.ru/file/263.html>

Требуется для работы со сканером отпечатков пальцев.

**Пример использования:**

#include <Adafruit\_Fingerprint.h>

Adafruit\_Fingerprint finger = Adafruit\_Fingerprint(&mySerial); – объявление объекта Finger. Параметр – ссылка на объектр для работы с UART, кокторому подключен модуль.

finger.begin();  – инициализация модуля отпечатков пальцев.

Func\_sensor\_communication(); – вызов модуля отпечатков пальцев.

Библиотеки коммуникации

Библиотека Wire

Скачать: <https://github.com/czukowski/Wire.h>

Требуется для работы с двухпроводным интерфейсом I2C.

**Пример использования:**

#include <Wire.h>

Wire.begin() – инициализация библиотеки, подключение к шине I2C.

Wire.requestFrom() – запрос мастером байтов от ведомого устройства.

Wire.beginTransmission() – начало передачи на ведомое устройство.

Библиотека Irremote

Скачать: <https://github.com/z3t0/Arduino-IRremote>

Требуется для работы ардуино с ИК приемником.

**Пример использования:**

#include <IRremote.h>

IRrecv irrecv(RECV\_PIN); – пин, к которому подключен ИК приемник.

SetPinAndButton(int ir1,int ir2,int pin) – позволяет настроить определенный выход на срабатывание при заданных значениях ir1, ir2.

Библиотека GSM

Скачать: <https://github.com/jt5/GSM>

Требуется для соединения через GSM-плату с сетью GSM/GRPS. С ее помощью можно реализовать операции, свершаемые GSM-телефоном, работать с голосовыми вызовами и подключаться к сети интернет через GRPS.

**Пример использования:**

#include <GSM.h>

GSM GSMAccess – инициализирует экземпляр класса.

gprs.powerOn() – включение питания.

GSMVoiceCall – настройка голосовых вызовов.

GPRS – настройка подключения к интернету.

GSM – управление радио-модемом.

Библиотека RFID

Скачать: <https://github.com/miguelbalboa/rfid>

Требуется для соединения Ардуино и [RFID](http://wikihandbk.com/ruwiki/index.php?title=RFID&action=edit&redlink=1)-модуля.

**Пример использования:**

#include <RFID.h>

RFID rfid(SS\_PIN, RST\_PIN); – создание экземпляра rfid, аргументы – пины, к которым подключен модуль.

rfid.init(); – инициализация модуля RFID.

Библиотека MFRC 522

Скачать: <https://github.com/miguelbalboa/rfid/archive/master.zip>

Требуется для соединения Ардуино и MFRC522 -модуля.

**Пример использования:**

#include <MFRC522.h>

MFRC522 mfrc522(SS\_PIN, RST\_PIN); – создание экземпляра MFRC522, аргументами указаны выходы, к которым подключен модуль.

mfrc522.PCD\_Init(); – инициализация MFRC522.

Библиотека Ethershield

Скачать: <https://github.com/thiseldo/EtherShield>

Новая версия https://github.com/jcw/ethercard

Требуется для подключения Ардуино к локальной сети или сети интернет. Библиотека больше не поддерживается, более новая версия Ethercard. Также существует стандартная библиотека Ethernet.

**Пример использования:**

#include «EtherShield.h»

#include <EtherCard.h>

EtherShield es = EtherShield (); – подготовка веб-страницы

ether.begin(sizeof Ethernet::buffer, mymac, ); – начало работы, аргументы – адрес Mac и номер порта, к которому подключен выход CS.

Библиотека Nrf24l01

Скачать: <https://github.com/maniacbug/RF24>

Требуется для работы с RF24-радиомодулем.

**Пример использования:**

#include “RF24.h”

RF24 – Конструктор создает новый экземпляр драйвера. Перед тем, как использовать, нужно создать экземпляр и указать пины, к которым подключен чип (\_cepin : контакт модуля Enable, cspin :  контакт модуля Select).

Begin – начало работы чипа.

setChannel – каналы для связи RF.

setPayloadSize – установка фиксированного размера передачи.

getPayloadSize – получение фиксированного размера.

Библиотека TinyGPS

Скачать: <https://github.com/mikalhart/TinyGPS>

Требуется для чтения сообщений GPGGA и GPRMC. Помогает считывать данные о положении, дате, времени, высоте и других параметрах.

**Пример использования:**

#include <TinyGPS.h>

TinyGPS gps; – создание экземпляра TinyGPS.

encode () – подача на объект последовательных данных по одному символу.

gps.stats( ) – метод статистики. Показывает, получены корректные данные или нет.

Библиотеки в Arduino IDE

Среди всего разнообразия библиотек можно выделить 3 основных группы:

* Встроенные – это библиотеки, изначально установленные в среде Arduino IDE. Их не нужно скачивать и устанавливать дополнительно, они доступны для использования в программе сразу после запуска среды разработки.
* Дополнительные – это библиотеки, которые нужно самостоятельно скачивать устанавливать. Обычно такой вид библиотек разрабатывает производитель датчиков, сенсоров и других компонентов для облегчения работы с ардуино.
* Зависимые библиотеки – устанавливаются как помощник дополнительной библиотеки, отдельно от нее не работает.

Самым простым способом работы с библиотеками в ардуино является использование встроенных возможностей среды разработки Arduino IDE. Об этом мы поговорим в отдельной статье.