

# **Начало работы с makegame**

С++ разработка  
программ на базе  
платформы makegame

Муниципальное учреждение дополнительного образования «Центр детского творчества №2»,  
2022-2023, Российская Федерация, город Саранск, ул. Грузинская 25

Документация по платформе makegame распространяется по лицензии Creative Commons Attribution-NoDerivatives 4.0 International (CC BY-ND).

Дата сборки: 26.11.2023

## Права использования

ТЕХНИЧЕСКИЕ ДАННЫЕ И ДАННЫЕ О НАДЕЖНОСТИ ПРОДУКТОВ MAKEGAME (ВКЛЮЧАЯ ТЕХНИЧЕСКИЕ ОПИСАНИЯ) С ИЗМЕНЕНИЯМИ, ВНОСИМЫМИ ВРЕМЯ ОТ ВРЕМЕНИ («РЕСУРСЫ»), ПРЕДОСТАВЛЯЮТСЯ ЦДТ2 («ЦДТ2») «КАК ЕСТЬ», И ЛЮБЫЕ ЯВНЫЕ ИЛИ ПОДРАЗУМЕВАЕМЫЕ ГАРАНТИИ, ВКЛЮЧАЯ, ПОМИМО ПРОЧЕГО, ПОДРАЗУМЕВАЕМЫЕ ГАРАНТИИ ТОВАРНОЙ ПРИГОДНОСТИ И ПРИГОДНОСТИ ДЛЯ ОПРЕДЕЛЕННОЙ ЦЕЛИ, НЕ ПРЕДОСТАВЛЯЮТСЯ. В МАКСИМАЛЬНОЙ СТЕПЕНИ, РАЗРЕШЕННОЙ ПРИМЕНИМЫМ ЗАКОНОДАТЕЛЬСТВОМ, ЦДТ2 НИ ПРИ КАКИХ ОБСТОЯТЕЛЬСТВАХ НЕ НЕСЕТ ОТВЕТСТВЕННОСТИ ЗА ЛЮБЫЕ ПРЯМЫЕ, КОСВЕННЫЕ, СЛУЧАЙНЫЕ, СПЕЦИАЛЬНЫЕ, ШТРАФНЫЕ ИЛИ КОСВЕННЫЕ УБЫТКИ (ВКЛЮЧАЯ, ПОМИМО ПРОЧЕГО, ПРИОБРЕТЕНИЕ ЗАМЕНЯЮЩИХ ТОВАРОВ ИЛИ УСЛУГ; ПОТЕРЯ ВОЗМОЖНОСТИ ИСПОЛЬЗОВАНИЯ, ДАННЫХ ИЛИ ПРИБЫЛИ; ИЛИ ПРЕРЫВАНИЕ ДЕЯТЕЛЬНОСТИ), НЕЗАВИСИМО ОТ ПРИЧИНЫ И НА ОСНОВАНИИ ЛЮБОЙ ТЕОРИИ ОТВЕТСТВЕННОСТИ, БУДЬ ТО ПО ДОГОВОРУ, СТРОГОЙ ОТВЕТСТВЕННОСТИ ИЛИ ДЕЛИКТУ (ВКЛЮЧАЯ НЕБРЕЖНОСТЬ ИЛИ ИНЫМ ОБРАЗОМ), ВОЗНИКАЮЩИМ КАКИМ-ЛИБО ОБРАЗОМ В РЕЗУЛЬТАТЕ ИСПОЛЬЗОВАНИЯ РЕСУРСОВ, ДАЖЕ ЕСЛИ БЫЛО СООБЩЕНО О ВОЗМОЖНОСТИ ТАКОГО УЩЕРБА

ЦДТ2 оставляет за собой право вносить любые улучшения, исправления или любые другие модификации в РЕСУРСЫ или любые продукты, описанные в них, в любое время и без дополнительного уведомления.

РЕСУРСЫ предназначены для опытных пользователей с соответствующим уровнем знаний в области цифрового дизайна. Пользователи несут полную ответственность за выбор и использование РЕСУРСОВ, а также за любое применение описанных в них продуктов. Пользователь соглашается возместить ущерб и оградить ЦДТ2 от всех обязательств, расходов, убытков или других убытков, возникающих в результате использования им РЕСУРСОВ. ЦДТ2 предоставляет пользователям разрешение на использование РЕСУРСОВ исключительно в сочетании с продуктами ЦДТ2. Любое другое использование РЕСУРСОВ запрещено. Никакая лицензия не предоставляется на какие-либо другие права интеллектуальной собственности ЦДТ2 или других третьих лиц.

## Об платформе makegame

Makegame — это недорогая, высокопроизводительная платформа для разработки it-решений на языке C++, с гибкими цифровыми интерфейсами, жидкокристаллическим дисплеем и аналоговыми стиками для управления.

Ключевые особенности:

- Микроконтроллер Pico 2040
- Два ядра Cortex M0+, до 133 МГц
- Интерфейсы UART, SPI, I2C
- SRAM 264 kB
- Flash память 2 МБ
- Монохромный дисплей с разрешением 128x64 пк., ST7565
- Два аналоговых стика для управления платформой
- Ядро Raspberry Pico 2040 автора @earlephilhower (<https://github.com/earlephilhower/arduino-pico>)
- Графическая библиотека U8g2 автора @olikraus (<https://github.com/olikraus/u8g2>)
- Библиотека makegame реализации методов платформы автора @avsavushkiny (<https://github.com/avsavushkiny/makegame>)

Мультизадачность	Pico 2040		Вывод данных на дисплей
	Dual core	up to 133MHz	
Планировщик задач, терминал	264kB	2MB	Подключение периферии через последовательные интерфейсы
	UART		
Аналоговые стики	ADC0...ADC3	SPI, I2C	

## Распределение портов, pin-out

Ключевые особенности использования микроконтроллера Pico 2040:

- Напряжение питания – 3.3V (Vss)
- Логический уровень – 3.3V
- 4 аналоговых порта ADC0...ADC3 (GP26, GP27, GP28, GP29)
- SPI интерфейс для вывода данных на дисплей ST7565 (GP16, GP17, GP18, GP19, GP20)
- Бuzzer (GP9)
- 2 аппаратных кнопки (GP6, GP7)
- Интерфейс I2C для периферии (GP4, GP5)
- Интерфейс SPI для периферии (GP0, GP1)
- Внутренний светодиод (GP25)

UART0 TX	GP0	PICO 2040	LED GP25	VBUS			GM612864-06D ST7565
UART0 RX	GP1			VSYS			
	GND			GND	VSS		
	GP2			3V3 EN			
	GP3			3V3 OUT	VDD_LCD, VDD_JOY		
I2C SDA	GP4			GP29	ADC3	JOY1X	
I2C SCL	GP5			GP28	ADC2	JOY1Y	
GND_KLCD	GND			GND			
JOY0_SW	GP6			GP27	ADC1	JOY0Y	
JOY1_SW	GP7			GP26	ADC0	JOY0X	
VDD_ALCD	GP8			RUN			
IN_BUZZER	GP9			GP22			
GND_BUZZER	GND			GND	JOYGND		
	GP10			GP21			
	GP11			GP20	RSE/RESET		
	GP12			GP19	SPI0 TX	SI/DATA	
	GP13			GP18	SPI0 SCK	SCL	
	GND			GND	GND	GND	
	GP14			GP17	SPI0 CSn	CS	
	GP15			GP16	SPI0 RX	RS/DC	

## Библиотека для разработки makegame (release 1.1)

Ключевые особенности библиотеки для разработки программ на C++ с использованием ядра Raspberry Pico 2040 @earlephilhower (<https://github.com/earlephilhower/arduino-pico>):

- Класс Joystick — управление платформой
- Класс Graphics — вывод данных на дисплей
- Класс Timer — запуск задач согласно временному интервалу – timer
- Класс Terminal — реализация запуска задач – terminal
- Класс Button, класс Shortcut, класс Cursor — визуальное оформление пользовательского интерфейса
- Класс Interface — реализация сообщений, диалогов

makegame	Joystick	calculatePositionX0	calculation of coordinates
		calculatePositionY0	
		calculatePositionX1	
		calculatePositionY1	
		calculatedIndexX0	index calculation
		calculatedIndexY0	
		calculatedIndexX1	
		calculatedIndexY1	
		pressKeyA	returns true/false
		pressKeyB	
		updatePositionXY	update sticks position
		posX0, posY0, posX1, posY1	sticks position
		OBJ_Y0, OBJ_X0, OBJ_Y1, OBJ_X1	counts objects, normally 0
	Graphics	controlBacklight	display backlight status
		initializationSystem	system setting
		print	display text
		render	displaying data
	Timer	timer	timer (millis)
	Terminal	terminal	task manager
	Button	button	visual button
	Shortcut	shortcut	visual shortcut
	Cursor	cursor	visual cursor
	Interface	message	message output
		greetingsBoard	greeting output

Версия библиотеки 1.1 (Release Version 1.1 · [avsavushkiny/makegame \(github.com\)](https://github.com/avsavushkiny/makegame))

makegame	Joystick	calculatePositionX0	returns values
		calculatePositionY0	
		calculatePositionX1	
		calculatePositionY1	
		calculatedIndexX0	returns values
		calculatedIndexY0	
		calculatedIndexX1	
		calculatedIndexY1	
	Graphics	pressKeyA	returns true/false
		pressKeyB	
		updatePositionXY	update sticks position
		posX0, posY0, posX1, posY1	value
		OBJ_Y0, OBJ_X0, OBJ_Y1, OBJ_X1	value
		controlBacklight	(bool state)
		initializationSystem	system setting
		print	(String text, int x, int y)
Timer	Timer	render	(void (*f)(), int timeDelay) (void (*f)())
		timer	(void (*f)(void), int interval)
	Terminal	terminal	multitasks
		button	(String text, uint8_t x, uint8_t y, void (*f)(void), int xCursor, int yCursor)
	Shortcut	shortcut	(const uint8_t *bitMap, uint8_t x, uint8_t y, void (*f)(void), int xCursor, int yCursor)
		cursor	(bool stateCursor, int xCursor, int yCursor)
	Interface	message	(String text, int duration)
		greetingsBoard	contains logo and text

## Примеры программ makegame (release 1.1)

### Hello world

```
#include <mg.h>

Graphics gfx;

void helloWorld()
{
    gfx.print("Hello world", 10, 10, 10, 3); // character spacing 3
    gfx.print("Hello world", 10, 20, 10, 4); // character spacing 4
    gfx.print("Hello world", 10, 30, 10, 5); // character spacing 5
    gfx.print("Hello world", 10, 40, 10, 6); // character spacing 6
    gfx.print("Hello world", 10, 50, 10, 7); // character spacing 7
    gfx.print("Hello world", 10, 60, 10, 8); // character spacing 8
}

void setup()
{
    gfx.initializationSystem();
}

void loop()
{
    gfx.render(helloWorld);
}
```

## Примеры программ makegame (release 1.1)

### Button

```
#include <mg.h>

Joystick joy; Graphics gfx; Interface inf; Button btn1, btn2; Cursor crs1;

void eventClickButton1()
{
    inf.message("Event Message", 1500);
}

void eventClickButton2()
{
    inf.message("Event Message", 1500);
}

void clickButtons()
{
    joy.updatePositionXY();
    crs1.cursor(true, joy.posX0, joy.posY0);

    gfx.print("Test Button", 10, 10, 8, 6);

    btn1.button("Button 1", 10, 25, eventClickButton1, joy.posX0, joy.posY0);
    btn2.button("Button 2", 60, 25, eventClickButton2, joy.posX0, joy.posY0);
}

void setup()
{
    gfx.initializationSystem();
}

void loop()
{
    gfx.render(clickButtons);
}
```



## Примеры программ makegame (release 1.1)

### Shortcut

```
#include <mg.h>

Joystick joy; Graphics gfx; Interface inf; Cursor crs1; Shortcut stc1;

const uint8_t sapper_bits[] = {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x01, 0x00,
    0x00, 0x40, 0x03, 0x00, 0x00, 0x40, 0x03, 0x00, 0x00, 0xF0, 0x0F, 0x00,
    0xC0, 0x0C, 0x30, 0x03, 0x60, 0x03, 0xC1, 0x06, 0xE0, 0x01, 0xC0, 0x07,
    0xC0, 0x90, 0x99, 0x03, 0x40, 0x20, 0x24, 0x03, 0x40, 0x91, 0x96, 0x03,
    0x20, 0x08, 0x6C, 0x07, 0x20, 0xCD, 0x99, 0x07, 0x38, 0x30, 0x66, 0x1F,
    0x24, 0x25, 0xD6, 0x27, 0xFC, 0x2A, 0xBF, 0x3F, 0xB8, 0xC1, 0xEF, 0x1F,
    0x60, 0x9F, 0xDF, 0x07, 0xE0, 0xAA, 0xEF, 0x07, 0xC0, 0x57, 0xF6, 0x03,
    0xC0, 0xAD, 0xFD, 0x03, 0x80, 0xB7, 0xFF, 0x01, 0x80, 0xFF, 0xFF, 0x01,
    0x40, 0xFF, 0xFF, 0x02, 0xC0, 0xFF, 0xFF, 0x03, 0x80, 0xF1, 0x8F, 0x01,
    0x00, 0xC0, 0x02, 0x00, 0x00, 0xC0, 0x02, 0x00, 0x00, 0x80, 0x01, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, };

void eventClickShortcut1()
{
    inf.message("Event Message", 1500);
}

void clickButtons()
{
    joy.updatePositionXY();

    gfx.print("Test Shortcut\nplease Click to Shortcut", 5, 10, 8, 5);
    stc1.shortcut(sapper_bits, 5, 30, eventClickShortcut1, joy.posX0, joy.posY0);
    crs1.cursor(true, joy.posX0, joy.posY0);
}

void setup()
{
    gfx.initializationSystem();
}

void loop()
{
    gfx.render(clickButtons);
}
```

---

**makegame**