KAMAMI

KAmod BlackPill 411

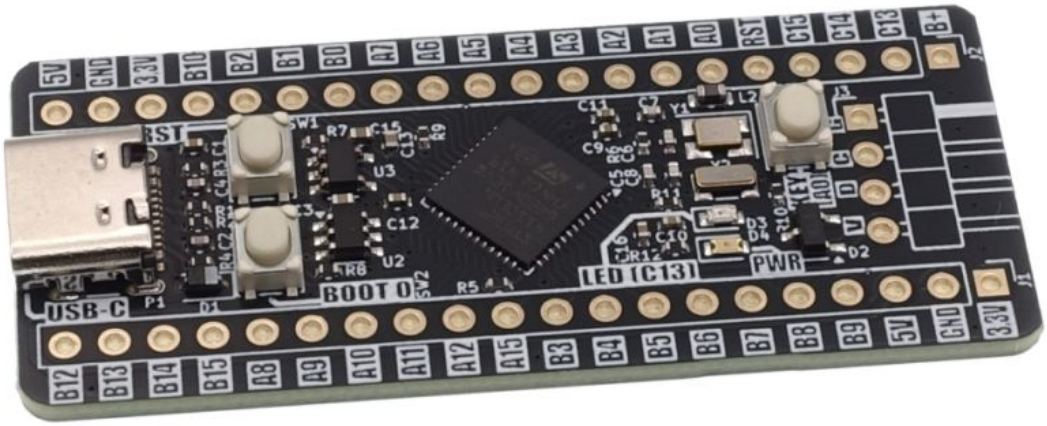
Rev. 20241125091732
Źródło: https://wiki.kamamilabs.com/index.php?title=KAmod_BlackPill_411

Table of contents

Description

Evaluation board with STM32F411CEU microcontroller, compatible with BlackPill

The **KAmod BlackPill 411** evaluation board contains the STM32F411CEU microcontroller and the components necessary to run and program it. The board is pin-compatible with the BlackPill project, but it is a completely new design, equipped with a USB-C connector with ESD protection and a modified power supply circuit. It can be programmed with Arduino IDE, because the system memory contains a suitable bootloader.

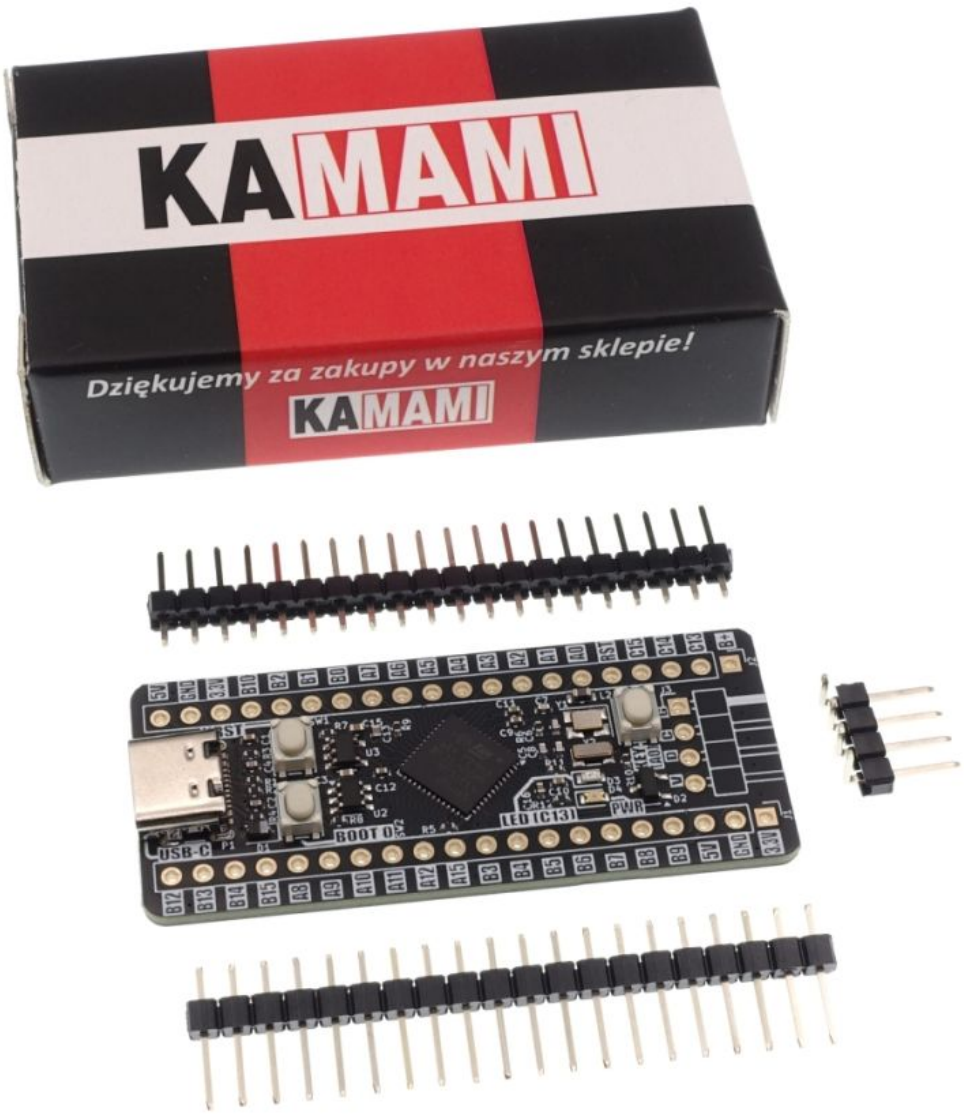


Basic features and parameters

- STM32F411CEU6 microcontroller: Cortex-M4, 512 kB Flash, 128 kB RAM, 100 MHz, 1 x 12-bit ADC, 7 timers, 3 x I2C, 2 x SPI, 3 x UART, USB OTG FS, RTC
- USB-C connector serves as a power connector, USB communication interface and allows for programming the microcontroller
- contains elements that filter interference and overvoltages on the USB interface lines
- 32 GPIO pins and 5 V and 3.3 V power lines available on standard 2.54 mm pitch connectors
- maximum load on the 5 V line is 500 mA, while for the 3.3 V line it is 200 mA
- precise resonator clocking the microcontroller and an independent resonator for the RTC module
- possibility of connecting a battery to support the module RTC
- SWD programming/debugging interface connector
- programmable via STM32CubeProgrammer and Arduino IDE
- board dimensions: 53.5x23 mm, height approx. 7 mm (without soldered goldpins)

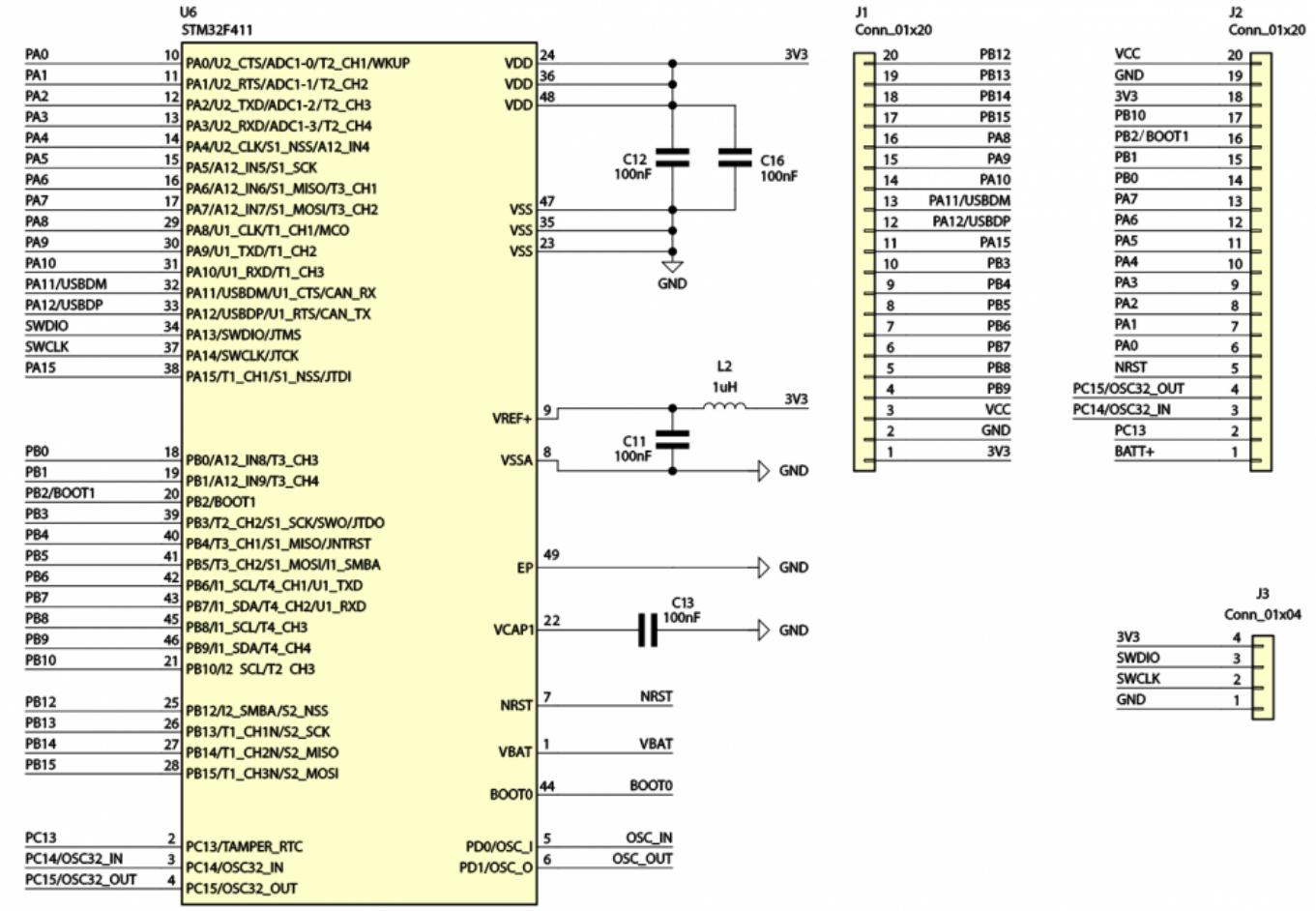
Standard equipment

Code	Description
KAmod BlackPill 411	• Assembled and started module, with bootloader uploaded • 2 x straight goldpin strip 20-pin raster 2.54 mm • 1 x angled goldpin strip 4-pin raster 2.54 mm

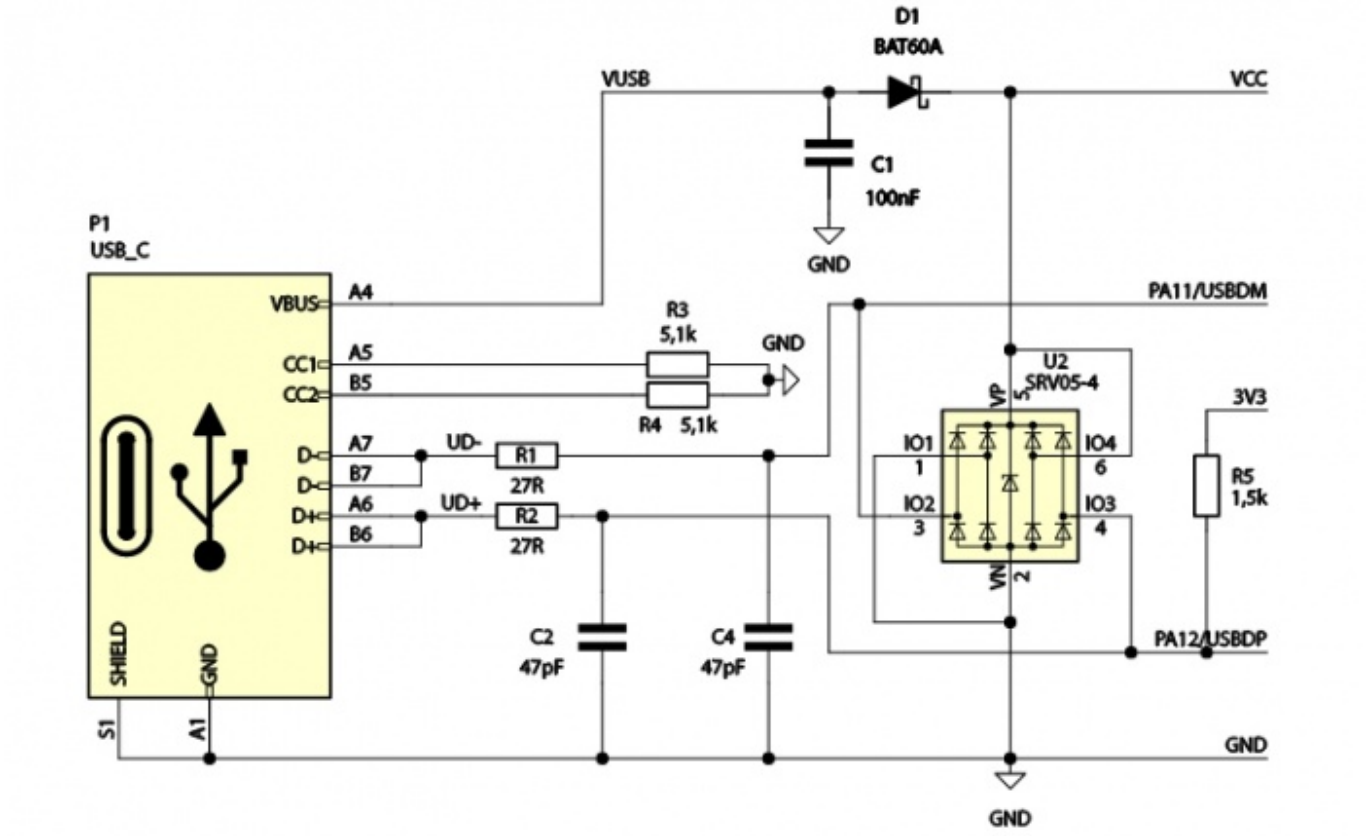


Electrical diagram

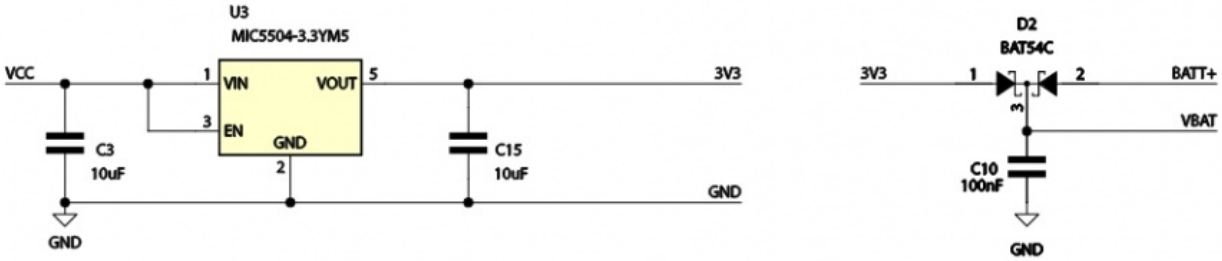
Microcontroller and GPIO and SWD connectors



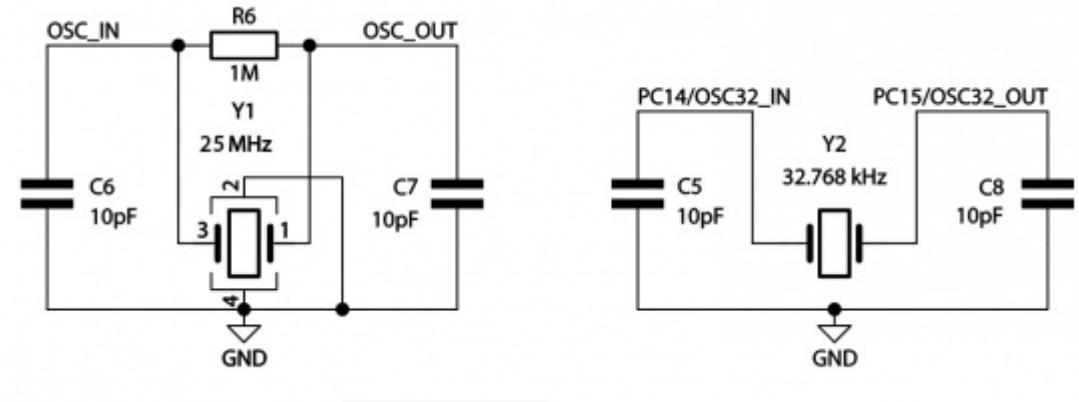
USB interface



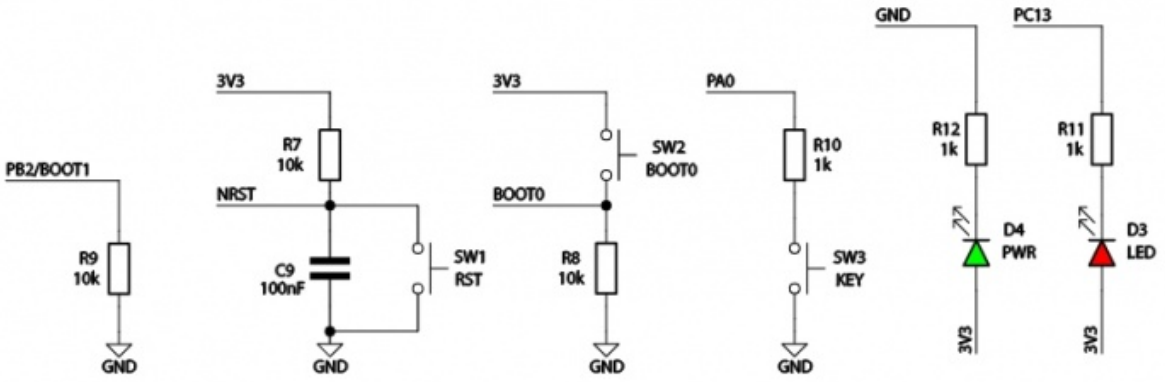
Power supply circuit



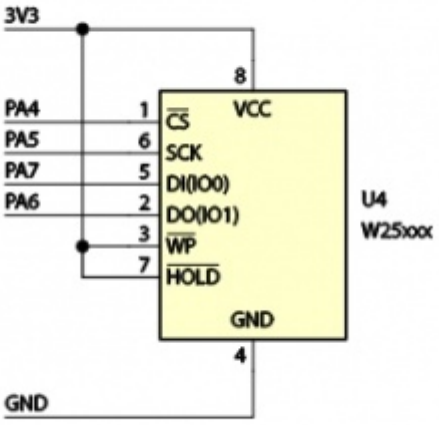
Microcontroller clock speed



Additional elements



External memory



Power

Connector	Function
USB-C **J1, J2**	• Supplies 5 V power to the module • Allows 5 V power and provides 3.3 V

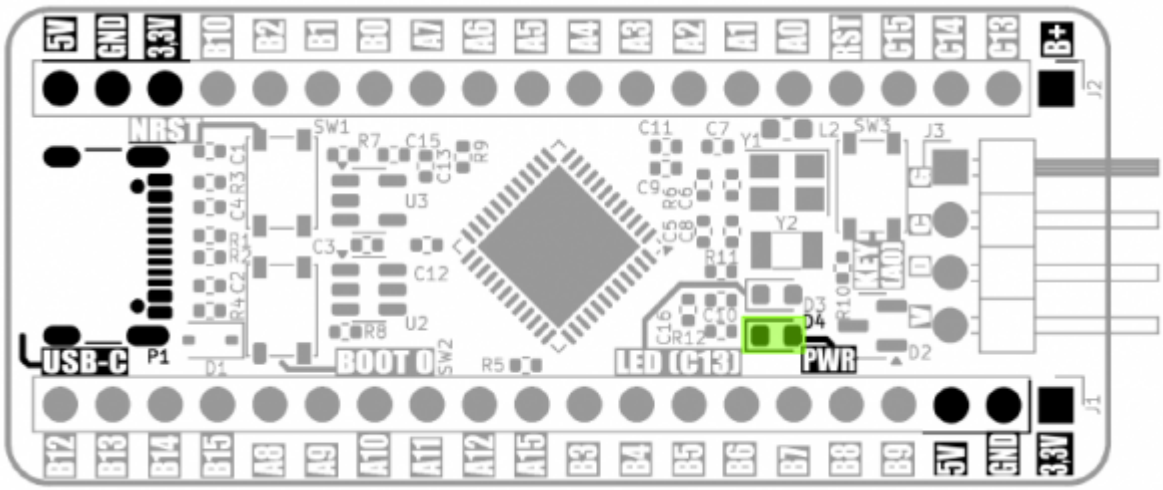
The **KAmod BlackPill 411** evaluation board can be powered in two ways:

- via the appropriate pins of the J1 and J2 connectors,
- via the USB-C connector.

A power source with a voltage in the range of 4.5...5.5 V and an efficiency of min. 100 mA should be connected to the contacts marked ' ***5V*** (plus) and ***GND*** (minus) on connectors J1 and/or J2. Then, on the contact marked ***3.3V***, a stabilized voltage of 3.3 V is available, which also powers the microcontroller. The presence of 3.3 V is indicated by the LED marked ***PWR***.

A standard USB power source with a capacity of at least 100 mA should be connected to the USB-C connector. Then, on the 5V contact of connector J1, a voltage of close to 5 V is available (relative to ground marked ***GND***). A small voltage drop (approx. 0.5 V) occurs on the Schottky diode, which allows current to flow in the direction from the USB-C connector to the board, but blocks current flow in the opposite direction - to the USB-C connector. Thanks to this, you can safely connect power in various configurations - USB and/or J1, J2 contacts.

There is a contact marked ***B+***' on the J2 connector. Together with the ground ***GND*** this is the power input from the battery supporting the RTC clock (integrated with the microcontroller). The backup battery voltage should be in the range of 1.65...3.6 V. Detailed information on the operation of the RTC module can be found in the STM32F411CEU microcontroller documentation.



USB Interface

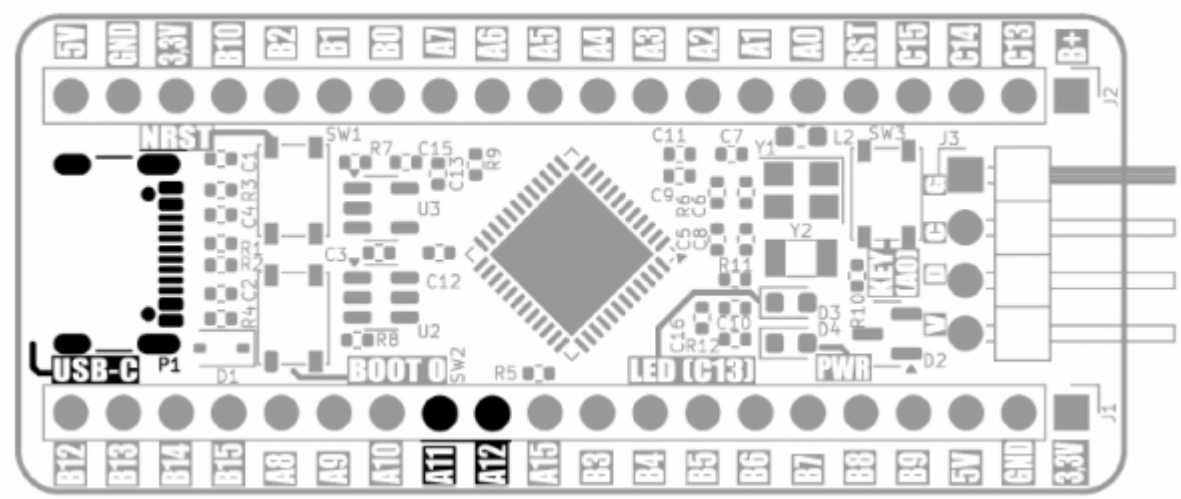
Connector	Function
USB-C	• Provides 5V power to the module • Can implement USB 2.0 FS Peripheral interface • Allows programming of microcontroller Flash memory

The USB-C connector is an easy way to provide power to the **KAmod BlackPill 411** board. In addition, the STM32F411CEU microcontroller has an integrated USB 2.0 FS interface controller, which can be used for communication, e.g. in VCP (*Virtual COM Port*) or CDC (*Communication Device Class*) mode.

The board has been configured in such a way that it allows the interface to operate in Peripheral mode, it is not prepared to work in Host (OTG) mode. The board also contains elements that filter possible interference and overvoltages on the USB interface lines, which ensure its stable operation.

An additional functionality of the USB-C connector is the ability to program the microcontroller's Flash memory. The appropriate bootloader is loaded into the microcontroller's non-volatile memory (ROM) at the production stage.

The USB interface lines (DP and DM) are simultaneously the GPIO PA12 and PA11 ports of the microcontroller. If we use USB, the contacts marked *A12* and *A11* cannot perform any other function - they must remain unconnected.



SWD programming/debugging interface

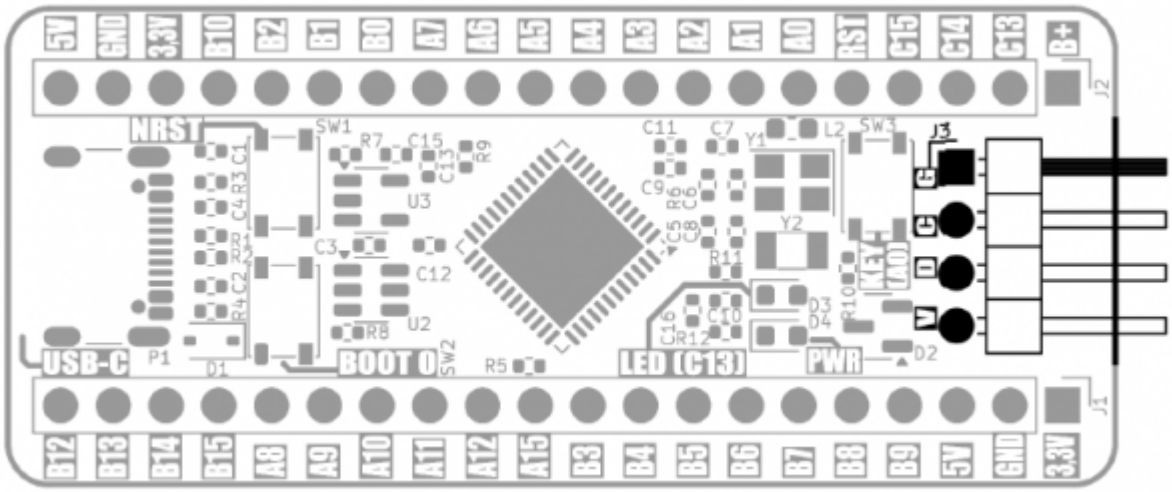
Connector	Function
J3	• Derived SWD (*Single Wire Debug*) interface with SWDIO and SWCLK signals

The SWD (*Single Wire Debug*) interface allows programming the microcontroller's Flash memory and monitoring the program's operation (debugging). Requires connecting an external programmer/debugger, e.g. STLINK-V2 or STLINK-V3MINIE.

The SWD interface was brought out to the J3 pin header. The signals were described as follows:

- G – system ground,
- C – SWCLK clock signal,
- D – SWDIO data signal,
- V – 3.3 V power supply line.

The signals should be connected to the same signals on the programmer/debugger connector. Sometimes SWCLK is also marked as TCK, while SWDIO is also marked as TMS. The programmer does not supply power to the **KAmod BlackPill 411** board, the power should be connected to the USB-C connector or J1/J2 pins.



Microcontroller Programming

Component	Function
SW1 – NRST	• Force microcontroller to zero state
SW2 – BOOT0	• Start microcontroller factory bootloader
J3 – SWD	• SWD (Single Wire Debug) programming interface
A9, A10 – UART	• UART (TXD, RXD) programming interface

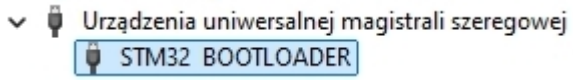
In a special area of the microcontroller's memory there is software that allows programming of its program memory - this is the so-called bootloader prepared by the microcontroller manufacturer. To start the bootloader, you must perform a specific sequence of actions, with the board's power supply connected:

1. Press and hold the NRTS button
2. Press and hold the BOOT 0 button while holding the NRST button
3. Release the NRTS button while holding the BOOT 0 button
4. Release the BOOT 0 button

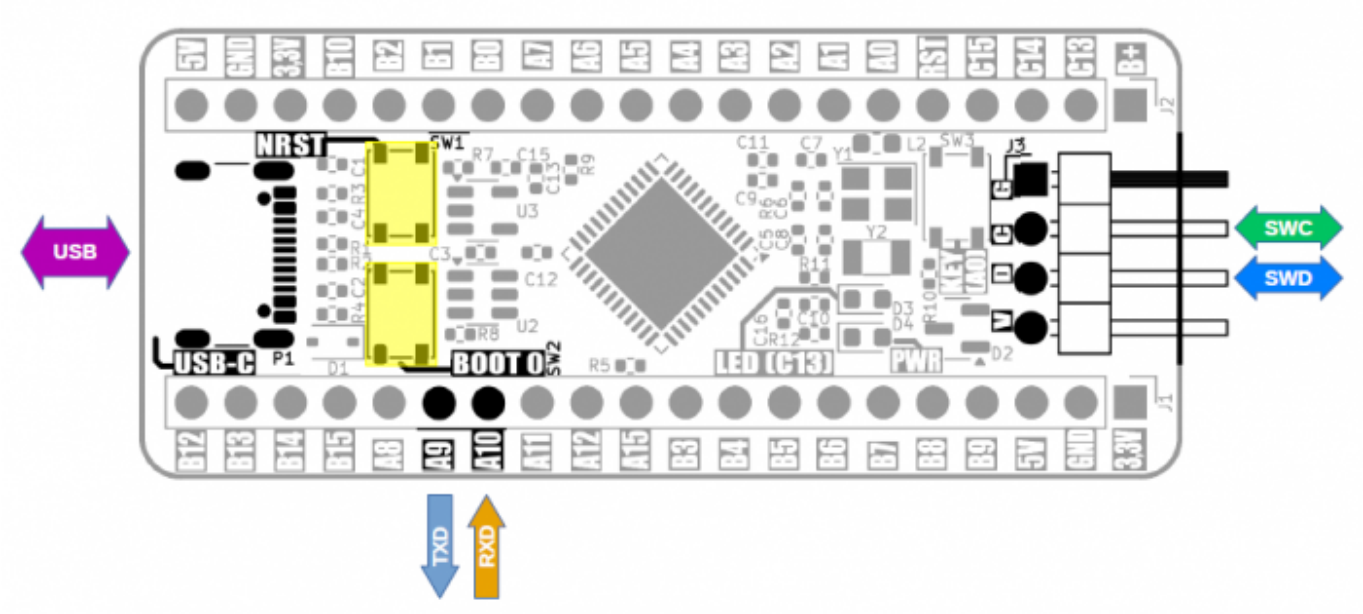
Another method of starting the bootloader is to connect the board's power supply while holding the BOOT 0 button.

Starting the bootloader activates two programming interfaces:

- **SWD**, which was described in the previous chapter and allows programming of the microcontroller using a programmer/debugger, e.g. STLINK-V2 or STLINK-V3MINIE.
- **UART**, which allows you to use a regular USB-UART converter as a programmer, e.g. KAmod USB-UART-mini. Then you need to connect the UART interface lines to the A9 and A10 pins, where:
  - A9 – the TXD output of the microcontroller should be connected to the RXD input of the USB-UART converter,
  - A10 – the RXD input of the microcontroller should be connected to the TXD output of the USB-UART converter.
- **USB**, which allows you to directly program the microcontroller connected to a PC. After the interface has been properly initialized, the microcontroller will be recognized as an STM32 BOOTLOADER device:



Due to certain technical conditions, the correct operation of the bootloader with the USB interface in the STM32F411CEU system is only possible when the system has a temperature close to 25°C. Only in such conditions is it possible to operate the integrated clock generator with sufficient precision. The microcontroller can be programmed via one of the interfaces: SWD, UART or USB using the STM32CubeProgrammer application, which is available on the official STMicroelectronics website: https://www.st.com/en/development-tools/stm32cubeprog.html

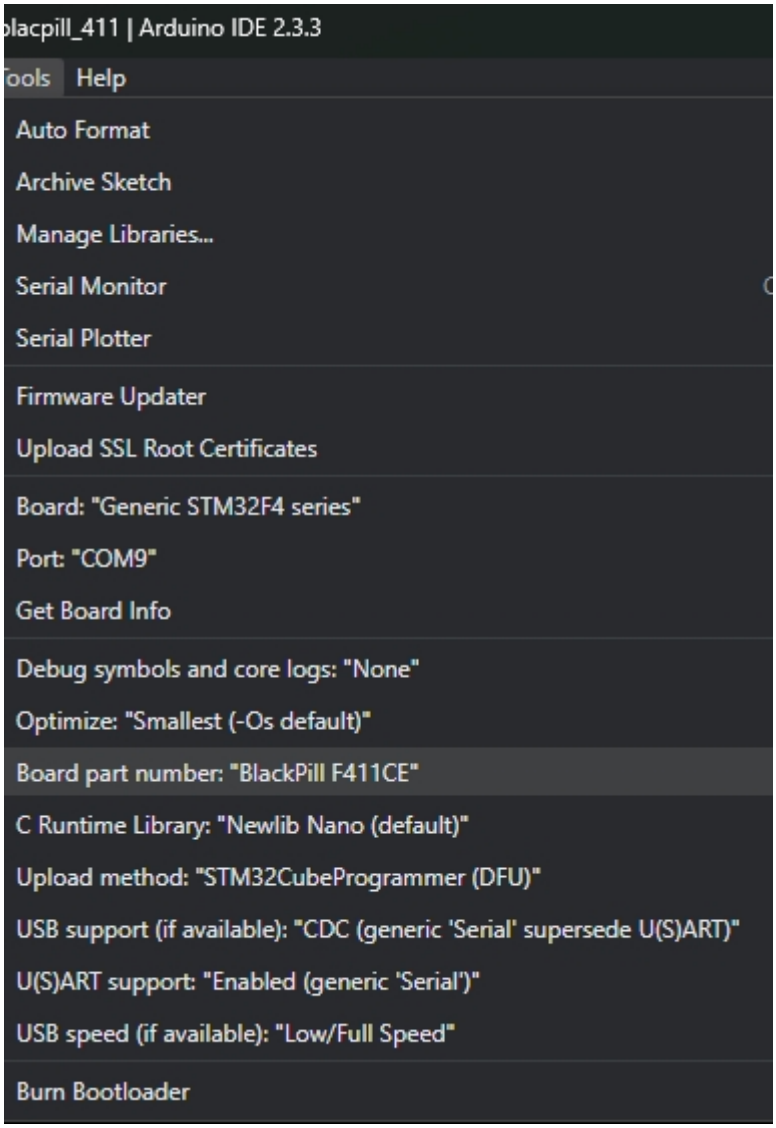


The **KAmod BlackPill 411** board allows programming the microcontroller via the USB interface using the Arduino IDE. Then you should:

- add the following address to the board manager sources (*Preferences*) in the Arduino IDE: https://github.com/stm32duino/BoardManagerFiles/raw/main/package_stmicroelectronics_index.json
- via the board manager (*Boards Manager*) install the package **STM32 MCU based boards**
- install STM32CubeProgrammer (https://www.st.com/en/development-tools/stm32cubeprog.html)

Then, after creating a new sketch, in the Tools tab:

- select the board: *BlackPill F411CC*,
- set *Upload method* to: *STM32CubeProgrammer(DFU)*,
- set *USB support* to: *CDC (generic Serial)*,
- *U(S)ART support* should be set to: *Enabled (generic Serial)*


lacpill_411 | Arduino IDE 2.3.3
ools Help
Auto Format
Archive Sketch
Manage Libraries...
Serial Monitor
Serial Plotter
Firmware Updater
Upload SSL Root Certificates
Board: "Generic STM32F4 series"
Port: "COM9"
Get Board Info
Debug symbols and core logs: "None"
Optimize: "Smallest (-Os default)"
Board part number: "BlackPill F411CE"
C Runtime Library: "Newlib Nano (default)"
Upload method: "STM32CubeProgrammer (DFU)"
USB support (if available): "CDC (generic 'Serial' supersede U(S)ART)"
U(S)ART support: "Enabled (generic 'Serial')"
USB speed (if available): "Low/Full Speed"
Burn Bootloader

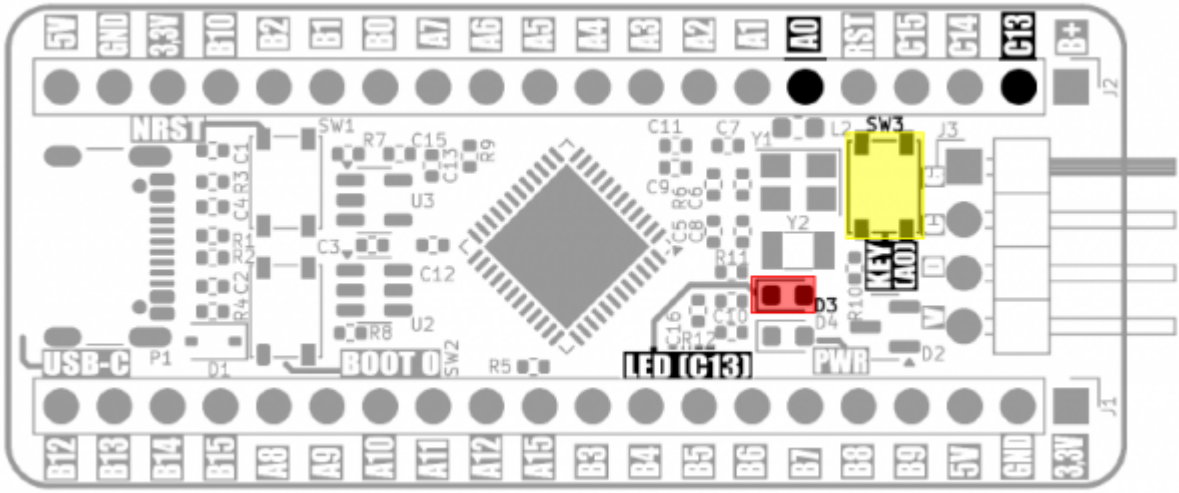
Additional elements – LED and button

Component	Function
D3 – LED (C13) **SW3 – KEY (A0)**	• LED connected to port PC13, active in L state • Micro button connected to port PA0, active in L state

The **KAmod BlackPill 411** board contains additional elements - an LED and a micro button, which can be used in the target application.

The LED is connected to port PC13, it lights up when the state is low.

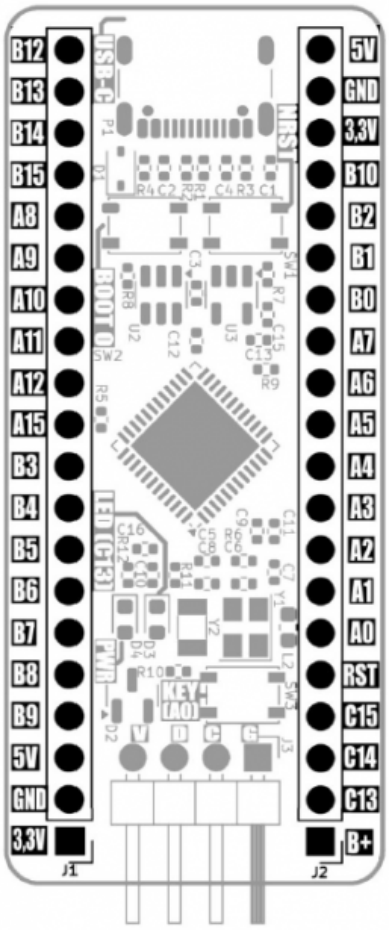
The micro button is connected to port PA0, pressing it forces a low state on this port. The button is connected via a 1k resistor, so there is no possibility of damaging port PA0 configured as an output.



GPIO connectors

Connector	Function
J1, J2	• Connectors with 20 pins with a 2.54 mm pitch, to which the GPIO ports of the microcontroller and 5 V and 3.3 V power supply are connected

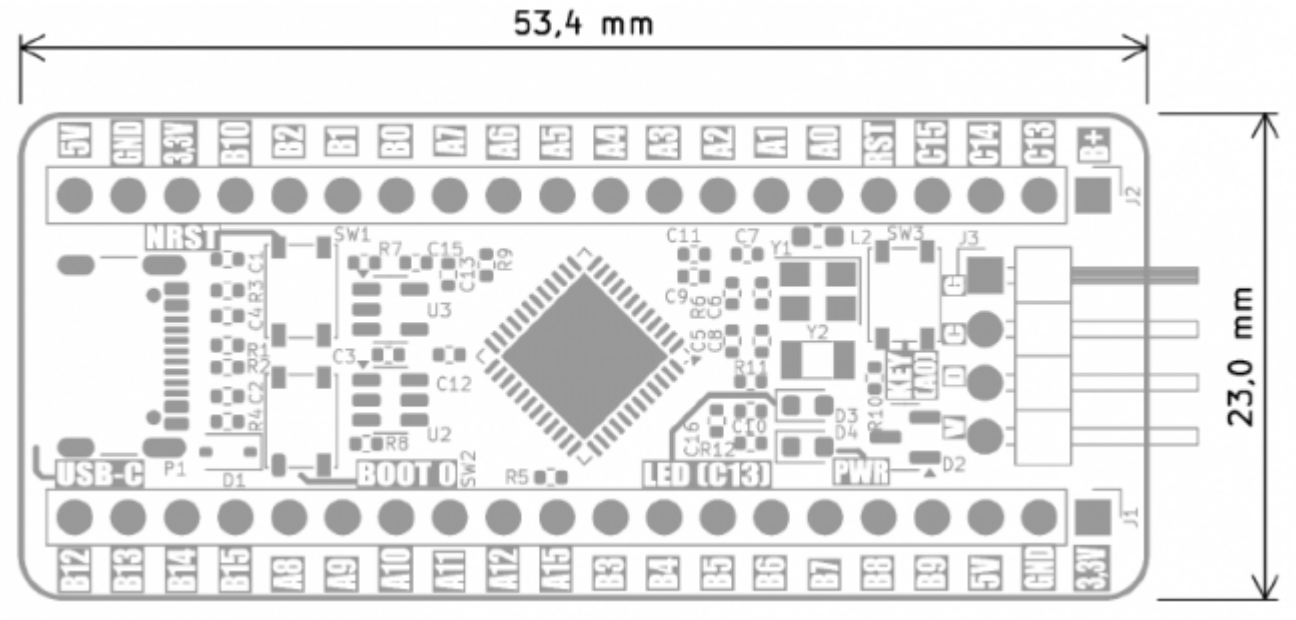
GPIO connectors J1 and J2 contain 20 pins each, to which the 5 V, 3.3 V power supply lines, GND and GPIO pins of the microcontroller module are connected. The GPIO ports are 5 V tolerant. The exact description of the pins and some of their additional functions are shown in the figure and table below.



J1		J2
PB12 (SPI2_NSS)		5 V
PB13 (SPI2_SCK)		GND
PB14 (SPI2_MISO)		3,3 V
PB15 (SPI2_MOSI)		PB10
PA8 (TIMER1_CH1)		(BOOT1) PB2
PA9 (UART1_TXD)		(ADC1_IN9) PB1
PA10 (UART1_RXD)		(ADC1_IN8) PB0
PA11 (USB_DM)		(ADC1_IN7) PA7
PA12 (USB_DP)		(ADC1_IN6) PA6
PA15 (SPI1_NSS)		(ADC1_IN5) PA5
PB3 (SPI1_SCK)		(ADC1_IN4) PA4
PB4 (SPI1_MISO)		(TIMER2_CH4) PA3
PB5 (SPI1_MOSI)		(TIMER2_CH3) PA2
PB6 (I2C1_SCL)		(TIMER2_CH2) PA1
PB7 (I2C1_SDA)		(TIMER2_CH1) PA0
PB8 (TIM4_CH3)		RESET
PB9 (TIM4_CH4)		(OSC_RTC_O) PC15
5 V		(OSC_RTC_I) PC14
GND		(LED_D3) PC13
3,3 V		RTC_BATTERY +

Dimensions

The dimensions of the **KAmod BlackPill 411**board are 53.5x23 mm, and the height is about 7 mm (without soldered goldpins).



Test program

The simplified code of the test program is below, it can be compiled in the Arduino environment.

```
#include <STM32RTC.h>

#define LED_PIN   PC13
#define SW_PIN    PA0

int message_period = 0;
int config_state = 0;
int ports_state = 0;
int pindex = 0;

STM32RTC& rtc = STM32RTC::getInstance();

void setup() {
  Serial.begin(115200);
  pinMode(LED_PIN, OUTPUT);
  pinMode(SW_PIN, INPUT_PULLUP);
  rtc.begin(); // initialize RTC 24H format
}

// the loop function runs over and over again forever
void loop() {
  //digitalWrite(LED_PIN, HIGH);
  if (digitalRead(SW_PIN) == HIGH){
    if ((ports_state&4) > 0){
      digitalWrite(LED_PIN, HIGH);
    } else {
      digitalWrite(LED_PIN, LOW);
    }

    if (config_state > 0){
      IO2input();
      config_state = 0;
    }

  } else {
    if ((ports_state&1) > 0){
      digitalWrite(LED_PIN, HIGH);
    } else {
      digitalWrite(LED_PIN, LOW);
    }
    if (config_state == 0){
      IO2output();
      config_state = 1;
    }


    if (config_state == 1){
      IO2low();
      IOwriteIndexed(pindex, 1);
      pindex++;
      if (pindex > 26) pindex = 1;
    }
  }
```

```
  // check if plugged into a host
  if (message_period >= 11){
    message_period = 0;
    int tmt = rtc.getMinutes();
    Serial.print("KAmod BlackPill411 RTC=");
    if (tmt < 10) Serial.print("0");
    Serial.print(tmt);
    tmt = rtc.getSeconds();
    Serial.print(":");
    if (tmt < 10) Serial.print("0");
    Serial.println(tmt);
  }
  delay(200);
  message_period++;
  ports_state++;
}

void IO2output(){
  //pinMode(PA0, OUTPUT); //switch
  pinMode(PA1, OUTPUT);
  pinMode(PA2, OUTPUT);
  pinMode(PA3, OUTPUT);
  pinMode(PA4, OUTPUT);
  pinMode(PA5, OUTPUT);
  pinMode(PA6, OUTPUT);
  pinMode(PA7, OUTPUT);

  pinMode(PA8, OUTPUT);
  pinMode(PA9, OUTPUT);
  pinMode(PA10, OUTPUT);
  pinMode(PA15, OUTPUT);

  pinMode(PB0, OUTPUT);
  pinMode(PB1, OUTPUT);
  pinMode(PB2, OUTPUT);
  pinMode(PB3, OUTPUT);
  pinMode(PB4, OUTPUT);
  pinMode(PB5, OUTPUT);
  pinMode(PB6, OUTPUT);
  pinMode(PB7, OUTPUT);
  pinMode(PB8, OUTPUT);
  pinMode(PB9, OUTPUT);

  pinMode(PB10, OUTPUT);

  pinMode(PB12, OUTPUT);
  pinMode(PB13, OUTPUT);
  pinMode(PB14, OUTPUT);
  pinMode(PB15, OUTPUT);
  //pinMode(PC13, OUTPUT);  //led
}

void IO2input(){
  //pinMode(PA0, INPUT); //switch
  pinMode(PA1, INPUT);
  pinMode(PA2, INPUT);
  pinMode(PA3, INPUT);
  pinMode(PA4, INPUT);
  pinMode(PA5, INPUT);
```

```
  pinMode(PA6, INPUT);
  pinMode(PA7, INPUT);

  pinMode(PA8, INPUT);
  pinMode(PA9, INPUT);
  pinMode(PA10, INPUT);
  pinMode(PA15, INPUT);

  pinMode(PB0, INPUT);
  pinMode(PB1, INPUT);
  pinMode(PB2, INPUT);

  pinMode(PB3, INPUT);
  pinMode(PB4, INPUT);
  pinMode(PB5, INPUT);
  pinMode(PB6, INPUT);
  pinMode(PB7, INPUT);
  pinMode(PB8, INPUT);
  pinMode(PB9, INPUT);

  pinMode(PB10, INPUT);

  pinMode(PB12, INPUT);
  pinMode(PB13, INPUT);
  pinMode(PB14, INPUT);
  pinMode(PB15, INPUT);

  //pinMode(PC13, INPUT);  //led
}

void IO2low(){
  //digitalWrite(PA0, LOW); //switch
  digitalWrite(PA1, LOW);
  digitalWrite(PA2, LOW);
  digitalWrite(PA3, LOW);
  digitalWrite(PA4, LOW);
  digitalWrite(PA5, LOW);
  digitalWrite(PA6, LOW);
  digitalWrite(PA7, LOW);

  digitalWrite(PA8, LOW);
  digitalWrite(PA9, LOW);
  digitalWrite(PA10, LOW);
  digitalWrite(PA15, LOW);

  digitalWrite(PB0, LOW);
  digitalWrite(PB1, LOW);
  digitalWrite(PB2, LOW);
  digitalWrite(PB3, LOW);
  digitalWrite(PB4, LOW);
  digitalWrite(PB5, LOW);
  digitalWrite(PB6, LOW);
  digitalWrite(PB7, LOW);
  digitalWrite(PB8, LOW);
  digitalWrite(PB9, LOW);

  digitalWrite(PB10, LOW);

  digitalWrite(PB12, LOW);
```

```
  digitalWrite(PB13, LOW);
  digitalWrite(PB14, LOW);
  digitalWrite(PB15, LOW);

  //digitalWrite(PC13, LOW);
}

void IO2high(){
  //digitalWrite(PA0, HIGH); //switch
  digitalWrite(PA1, HIGH);
  digitalWrite(PA2, HIGH);
  digitalWrite(PA3, HIGH);
  digitalWrite(PA4, HIGH);
  digitalWrite(PA5, HIGH);
  digitalWrite(PA6, HIGH);
  digitalWrite(PA7, HIGH);

  digitalWrite(PA8, HIGH);
  digitalWrite(PA9, HIGH);
  digitalWrite(PA10, HIGH);
  digitalWrite(PA15, HIGH);

  digitalWrite(PB0, HIGH);
  digitalWrite(PB1, HIGH);
  digitalWrite(PB2, HIGH);
  digitalWrite(PB3, HIGH);
  digitalWrite(PB4, HIGH);
  digitalWrite(PB5, HIGH);
  digitalWrite(PB6, HIGH);
  digitalWrite(PB7, HIGH);
  digitalWrite(PB8, HIGH);
  digitalWrite(PB9, HIGH);

  digitalWrite(PB10, HIGH);

  digitalWrite(PB12, HIGH);
  digitalWrite(PB13, HIGH);
  digitalWrite(PB14, HIGH);
  digitalWrite(PB15, HIGH);

  //digitalWrite(PC13, HIGH);
}

void IOwriteIndexed(int index, int state){
  int pp = 1;

  switch (index){
    //case 0: pp = PA0; break;
    case 1: pp = PA1; break;
    case 2: pp = PA2; break;
    case 3: pp = PA3; break;
    case 4: pp = PA4; break;
    case 5: pp = PA5; break;
    case 6: pp = PA6; break;
    case 7: pp = PA7; break;

    case 8: pp = PB0; break;
    case 9: pp = PB1; break;
    case 10: pp = PB2; break;
```

```
    case 11: pp = PB10; break;

    case 12: pp = PB12; break;
    case 13: pp = PB13; break;
    case 14: pp = PB14; break;
    case 15: pp = PB15; break;

    case 16: pp = PA8; break;
    case 17: pp = PA9; break;
    case 18: pp = PA10; break;
    //case 19: pp = PA11; break
    //case 20: pp = PA12; break
    case 19: pp = PA15; break;

    case 20: pp = PB3; break;
    case 21: pp = PB4; break;
    case 22: pp = PB5; break;
    case 23: pp = PB6; break;

    case 24: pp = PB7; break;
    case 25: pp = PB8; break;
    case 26: pp = PB9; break;
  }

  pinMode(pp, OUTPUT);
  if (state > 0){
    digitalWrite(pp, HIGH);
  } else {
    digitalWrite(pp, LOW);
  }
}
```

The program works by starting a serial port based on the USB interface and the RTC clock module and cyclically sending time information through this serial port.

The program is accompanied by a slow flashing of the D3 LED. When the KEY (SW3) button is pressed, the high state will be set on all ports in turn and the LED will start flashing quickly.

Links

- CAD Model (STEP)
- Data sheet of the STM32F411CEU system
- Arduino test program
- Test project STM32CUBE_IDE
- STLINK-V3MINIE - compact programmer/debugger for STM32
- STM32CubeProgrammer
- KAmod USB-UART-mini - Miniature USB-UART converter

BTC Korporacja
05-120 Legionowo
ul. Lwowska 5
tel.: (22) 767-36-20
faks: (22) 767-36-33
e-mail:
sprzedaz@kamami.pl
https://kamami.pl