

B0B99PRPA

10. přednáška - Nucleo, MBED

Obsah přednášky

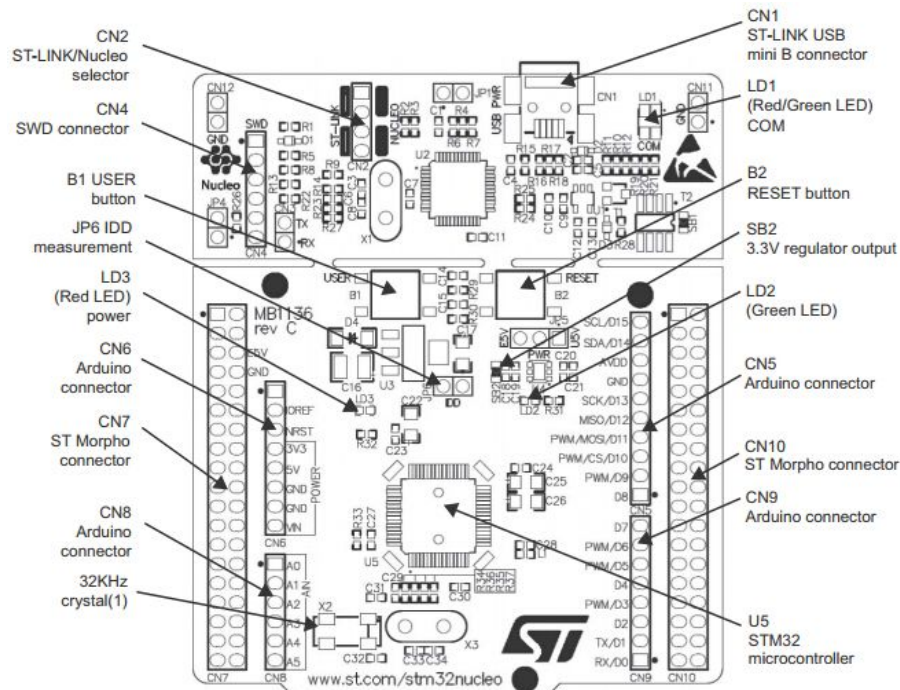
- Vývojová deska Nucleo F446RE
- HW komponenty, přerušení
- Nastavení offline vývojového prostředí
- SW komponenty

Vývojová deska Nucleo F446RE

- Mikrokontrolér STM32F446RE
 - ARM®32-bit Cortex®-M4 CPU
 - 180 MHz max CPU frequency
- Paměť
 - 512 KB Flash
 - 128 KB SRAM System
 - 4 KB SRAM Backup
- GPIO (50), externí přerušení
- 12-bit ADC (3) - 16 kanálů
- 12-bit DAC - 2 kanály
- USART (4)
- UART (2)
- SPI (4)
- CAN (2)
- SPDIF-Rx (1)
- HDMI-CEC (1)
- Quad SPI (1)
- Camera Interface
- ...

Featurky

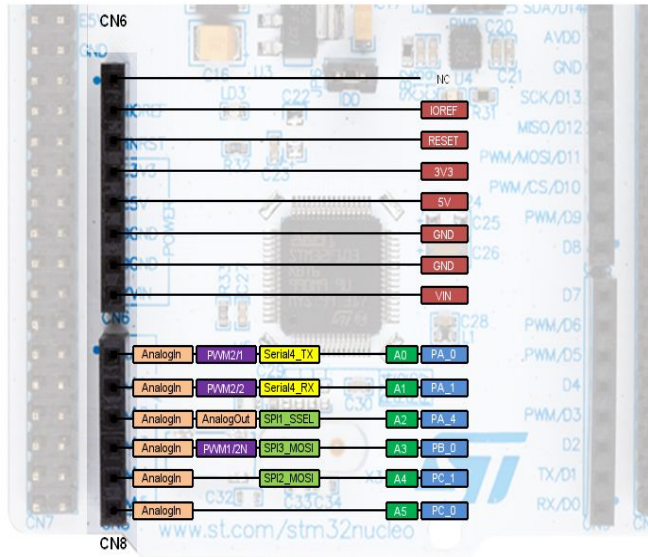
- ST-LINK/V2-1
 - Integrovaný debugger/programátor
- Arduino kompatibilní interface
- Flexibilní možnosti napájení
 - USB VBUS
 - externí zdroj (3.3 V, 5 V, 7 - 12 V)
- Tři rozdílná USB rozhraní
 - virtuální COM port
 - Mass Storage
 - debug port



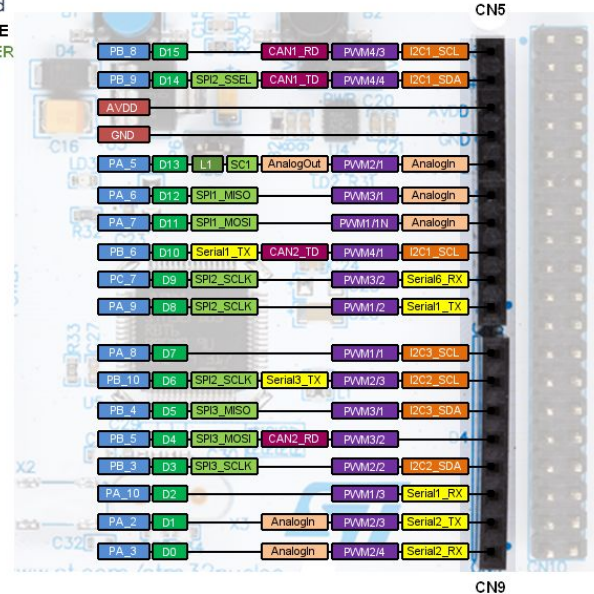
Arduino kompatibilní rozhraní



life.augmented
NUCLEO-F446RE
ARDUINO HEADER
(top left side)

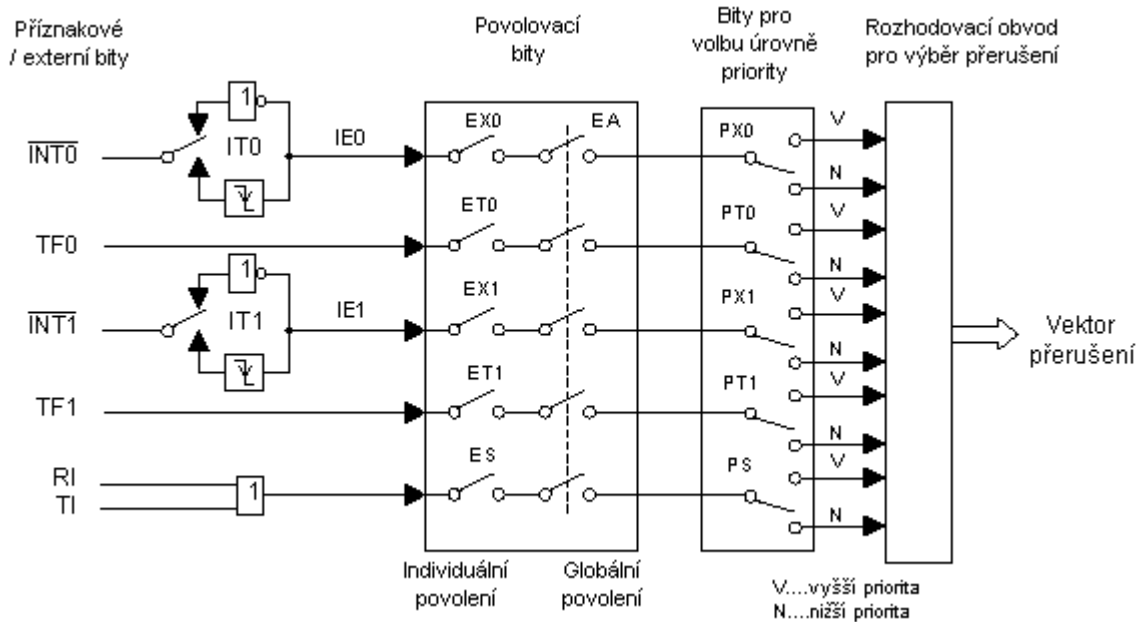


life.augmented
NUCLEO-F446RE
ARDUINO HEADER
(top right side)



HW mikrokontroléru

Přerušovací systém mikrokontroléru

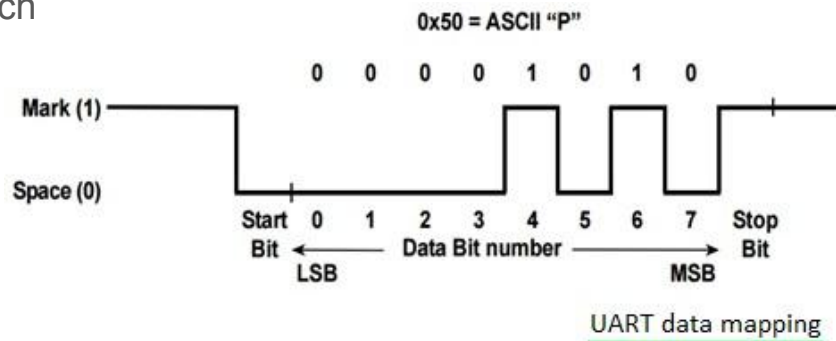


Zdroje přerušení

- Interní
 - čítače/časovače
- Externí
 - U(S)ART
 - GPIO

Sériová komunikace

- Universal Asynchronous Receiver/Transmitter (UART, RS232)
 - přijímač musí znát přenosovou rychlost
 - jednoduchý protokol, přenos po bytech
 - start bit
 - datové slovo (payload)
 - stop bit (jeden nebo dva)
 - paritní bit



- Universal Synchronous/Asynchronous Receiver/Transmitter (USART)
 - generuje hodinový signál, není třeba znát přenosovou rychlost
 - přenos dat po blocích
 - může být použit jako UART

Debugování

Možnosti debugování

Instrumentace

- kontrolní výpisy na sériovou linku

On-chip debugging

- Open OCD + GDB

Nastavení sériového portu

- virtuální COM port / GPIO
- inicializace

```
Serial pc(SERIAL_TX, SERIAL_RX);
```

```
Serial pc(USBTX, USBRX);
```

- parametrizace
 - defaultní nastavení: 9600-8-N-1
- přerušení

Nastavení sériového portu

Public Member Functions

Serial (PinName tx, PinName rx, const char *name=NULL)

Create a **Serial** port, connected to the specified transmit and receive pins.

void **baud** (int baudrate)

Set the baud rate of the serial port.

void **format** (int bits=8, Parity parity=SerialBase::None, int stop_bits=1)

Set the transmission format used by the serial port.

int **readable** ()

Determine if there is a character available to read.

int **writeable** ()

Determine if there is space available to write a character.

void **attach** (void(*fptr)(void), IrqType type=RxIrq)

Attach a function to call whenever a serial interrupt is generated.

template<typename T >

void **attach** (T *tptr, void(T::*mptr)(void), IrqType type=RxIrq)

Attach a member function to call whenever a serial interrupt is generated.

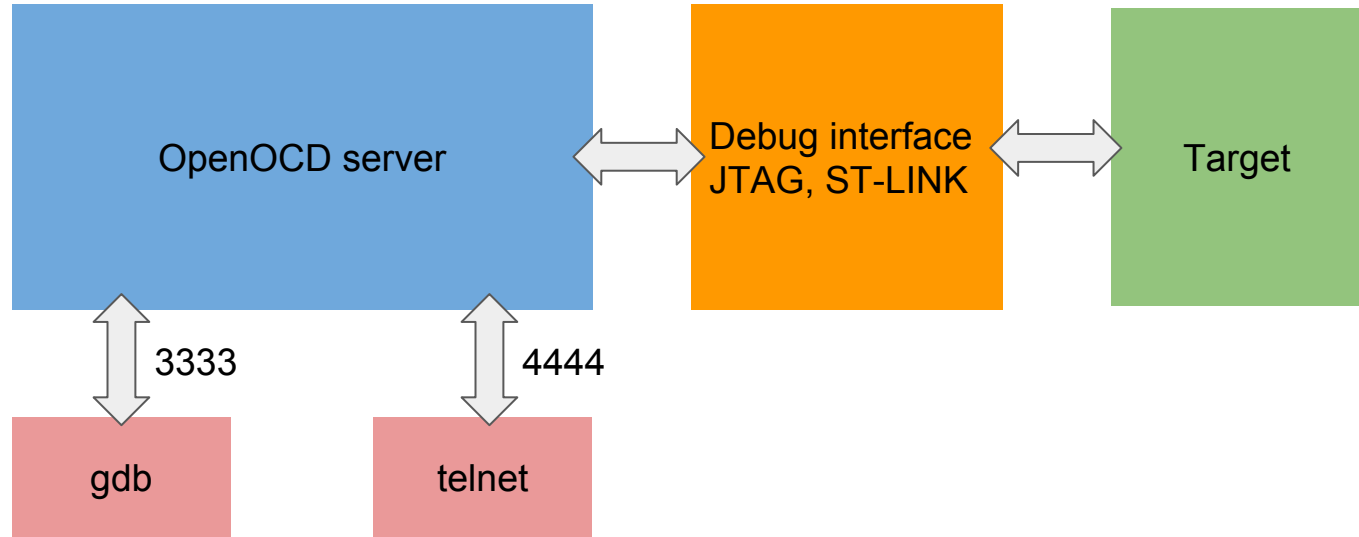
void **send_break** ()

Generate a break condition on the serial line.

void **set_flow_control** (Flow type, PinName flow1=NC, PinName flow2=NC)

Set the flow control type on the serial port.

OpenOCD



DigitalIn - digitální vstup

Deklaruje GPIO pin jako vstupní

```
DigitalIn name(Pin);
```

```
DigitalIn mybutton(USER_BUTTON);
```

```
DigitalIn IN1(PA_10);
```

```
name.mode(Pull_Mode);
```

Mód: PullUp, PullDown, PullNone

Public Member Functions

DigitalIn (PinName pin)
Create a **DigitalIn** connected to the specified pin.

DigitalIn (PinName pin, PinMode mode)
Create a **DigitalIn** connected to the specified pin.

int **read** ()
Read the input, represented as 0 or 1 (int)

void **mode** (PinMode pull)
Set the input pin mode.

int **is_connected** ()
Return the output setting, represented as 0 or 1 (int)

operator int ()
An operator shorthand for **read()**

InterruptIn - ext. přerušení

- Konfigurace GPIO pinu jako ext. zdroje přerušení

```
#include "mbed.h"
InterruptIn event(p16);

DigitalOut led(LED1);

void trigger() {
    printf("triggered!\n");
}

int main() {
    event.rise(&trigger);
    while(1) {
        led = !led;
        wait(0.25);
    }
}
```

	<code>InterruptIn</code> (PinName pin)
int	<code>read ()</code>
	<code>operator int ()</code>
void	<code>rise (Callback< void()> func)</code>
	<code>template<typename T , typename M ></code>
void	<code>rise (T *obj, M method)</code>
void	<code>fall (Callback< void()> func)</code>
	<code>template<typename T , typename M ></code>
void	<code>fall (T *obj, M method)</code>
void	<code>mode (PinMode pull)</code>
void	<code>enable_irq ()</code>
void	<code>disable_irq ()</code>

DigitalOut - digitální výstup

- Definuje GPIO pin jako výstupní
 - je možné přiřadit hodnotu 0 nebo 1

```
DigitalOut Name(Pin);
```

```
DigitalOut Rele(PA_8);
```

```
Rele = 1;
```

```
Rele = !Rele;
```

Public Member Functions

	DigitalOut (PinName pin) Create a DigitalOut connected to the specified pin.
	DigitalOut (PinName pin, int value) Create a DigitalOut connected to the specified pin.
void	write (int value) Set the output, specified as 0 or 1 (int)
int	read () Return the output setting, represented as 0 or 1 (int)
int	is_connected () Return the output setting, represented as 0 or 1 (int)
DigitalOut &	operator= (int value) A shorthand for write()
	operator int () A shorthand for read()

Ticker, Timer, Timeout

- Ticker
 - cyklické spuštění callback funkce
 - <https://os.mbed.com/docs/latest/reference/ticker.html>
- Timer
 - vytvoření, nastartování, zastavení a čtení časovače
 - <https://os.mbed.com/docs/latest/reference/timer.html>
- Timeout
 - jednorázové spuštění callback funkce po uplynutí intervalu
 - <https://os.mbed.com/docs/latest/reference/timeout.html>

To je zatím všechno...