

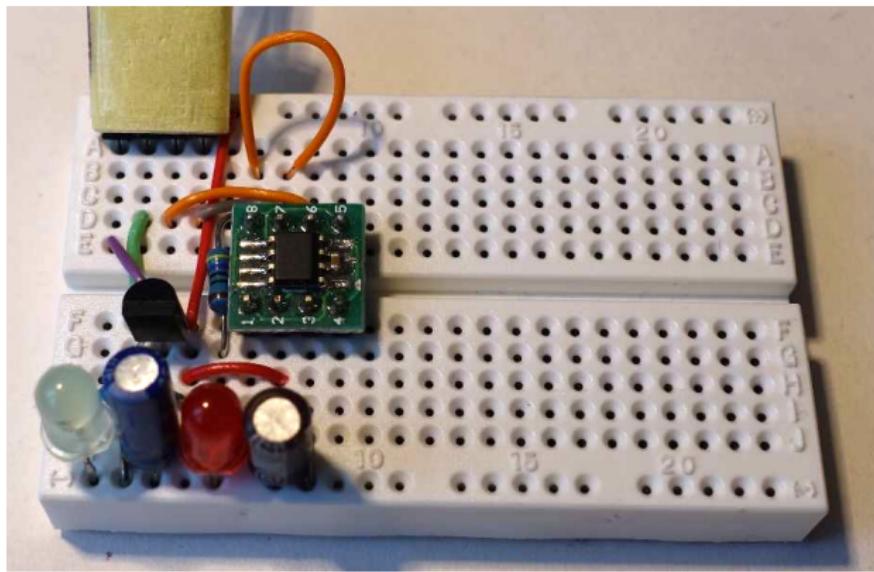
# Úvod do programovaní mikrokontrolérů STM32

Stanislav Novák

3. dubna 2023

# Úvod, cíl přednášky

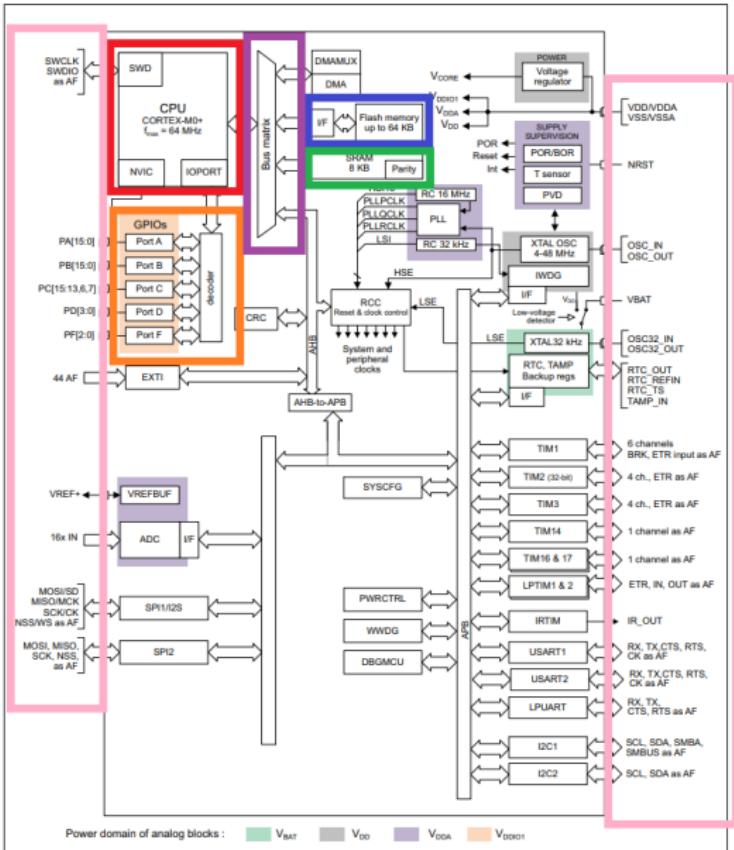
- **cíl** = zkusit si programovaní mikrokontroléru



- Bude programován čip "oscíloskopu" z předchozích setkání.

# Formulace problému, hardware

- **program** = posloupnost instrukcí řešící *nějakou úlohu*
- **mikrokontrolér** = integrovaný obvod počítače (obsahuje **CPU**, **paměti**, **i/o**, ...)



- **binární program** (de facto vždy); obrázek vpravo
- **kompilovaný program**, obvykle C/C++
  - CMSIS drivery
  - **ST HAL/LL**
  - ARM mbed
  - **Arduino; Ardublock**
  - další
- **interpretovaný program** bez komplikace, program běží v programu (interpretru)  
**MicroPython**

Address	0	4	8	C
0x00000000	20001EF4	080007B7	080007B5	080007C5
0x00000010	080007C7	080007C9	080007CB	00000000
0x00000020	00000000	00000000	00000000	080007CD
0x00000030	080007CF	00000000	080007D1	0800086D
0x00000040	080014CB	080014CB	080014CB	080014CB
0x00000050	080014CB	080014CB	080014CB	080014CB
0x00000060	080014CB	080014A1	080014AB	080014BB
0x00000070	080014CB	080014CB	080014CB	080014CB

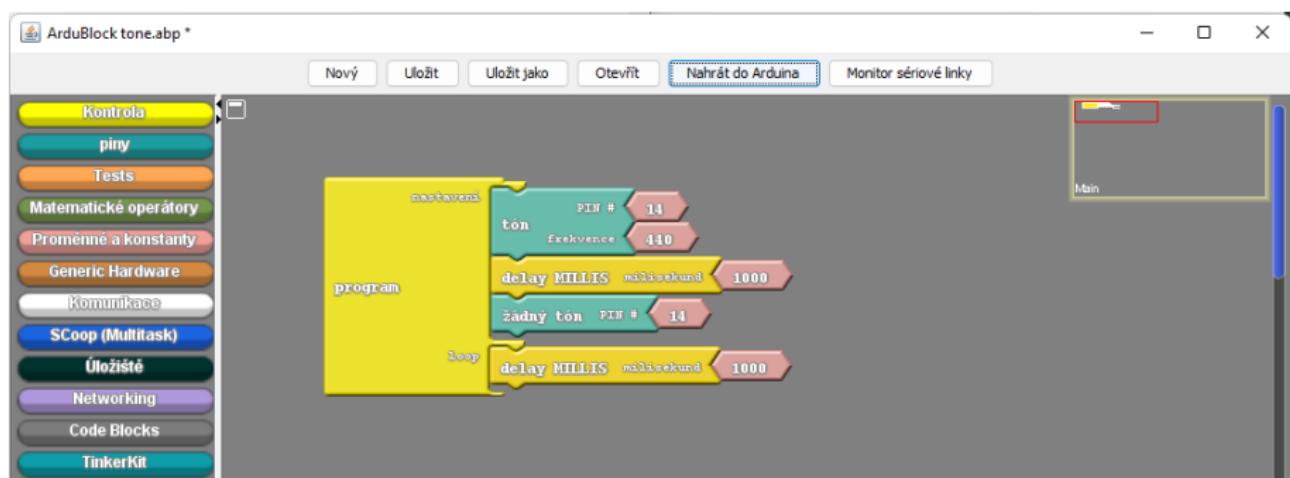
```
.\stm32g030j6.bin:      file format binary
```

```
Disassembly of section .data:
```

```
00000000 <.data>:  
 0:    1ef4        subs   r4, r6, #3  
 2:    2000        movs   r0, #0  
 4:    07b7        lsls   r7, r6, #30  
 6:    0800        lsr s r0, r0, #32  
 8:    07b5        lsls   r5, r6, #30  
 a:    0800        lsr s r0, r0, #32  
 c:    07c5        lsls   r5, r0, #31  
 e:    0800        lsr s r0, r0, #32  
 10:   07c7        lsls   r7, r0, #31  
 12:   0800        lsr s r0, r0, #32  
 14:   07c9        lsls   r1, r1, #31  
 16:   0800        lsr s r0, r0, #32  
 18:   07cb        lsls   r3, r1, #31  
 1a:   0800        lsr s r0, r0, #32  
      ...  
 2c:   07cd        lsls   r5, r1, #31  
 2e:   0800        lsr s r0, r0, #32  
 30:   07cf        lsls   r7, r1, #31
```

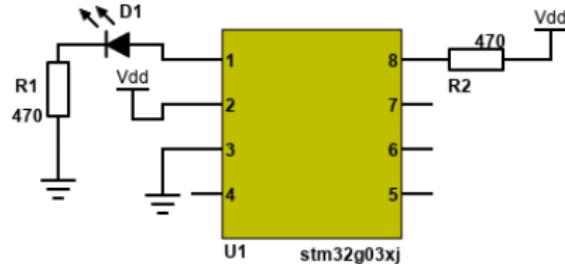
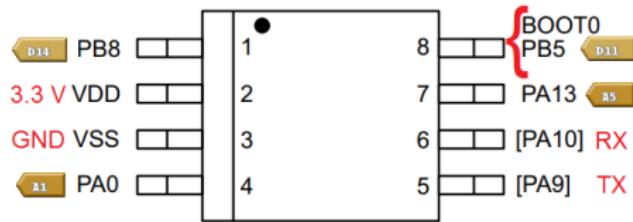
# Arduino + stm32duino + Ardublock

- Arduino je současným „kutilským standardem“
- použijí se dva rozšiřující balíky STM32duino (STM32) + Ardublock (programovaní pomocí „bločků“)
- výsledek = možnost začít relativně rychle programovat (=řešit *nějaké úlohy*)



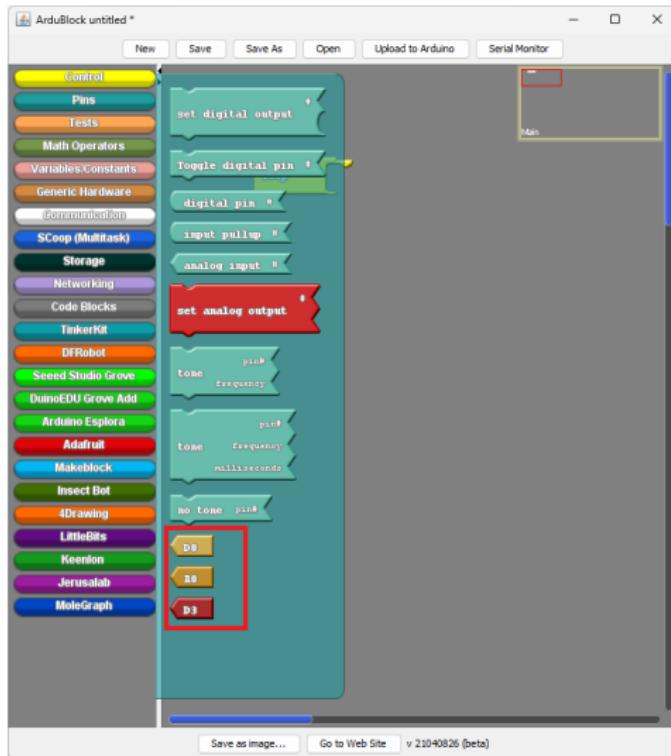
# Ardublock ukázka

- (v přednášce následuje zprovoznění ukázka prvního programu)
- stm32duino ve svých definicích přemapovává piny arduina (A0-Ax, D0-Dx) na piny stm32 (Pxy, např PA0); zde postačí diagram na obrázku (vlevo)
- k programovaní využijeme tzv. bootloader; upravíme naše zapojení podle obrázku (vpravo)



# Omezení Ardublocku

- Arduino rozlišuje piny digitální, analogové a PWM, tomu odpovídají barvy bloků pinů Ardublocku (obrázek)
- Ardublock postrádá "dopřednou kompatibilitu" s piny STM32
- → používejte/hledejte piny nezávislé na kategorii
- Analogové vstupy 4, 7
- PWM výstupy 1, 4, 8



# Nahrání osciloskopu

Při nahrání programu jsem přepsali program osciloskopu. Osciloskop můžeme znovu nahrát pomocí programu **STM32CubeProg** (postup v slidech ke cvičení, ukázka na přednášce).

