

Základní nastavení pro Mbed:

Kurzívou jsou psané proměnné, nastavení, apod., které mění programátor. Např. umístění vstupů, výstupů, proměnných.

Nastavení pinu jako výstup:

```
DigitalOut output(umisteni); // DigitalOut led(A2);
```

Nastavení pinu jako vstup, vč. nastavení chování:

```
DigitalIn input(umisteni); // DigitalIn button(D2);  
input.mode(mode); // button.mode(PullUp);
```

Možné nastavení módů:

- PullUp - Nastavení vstupu s měkkou logickou jedničkou
- PullDown - Nastavení vstupu s měkkou logickou nulou

Časová prodleva:

```
wait(cas v s); // 1.2 vteřiny = wait(1.2);
```

I²C sběrnice

1. Definice I²C sběrnice pomocí (globálně)

```
I2C i2c(I2C_SDA, I2C_SCL);
```

2. Zápis/čtení dat se provede pomocí příkazů

```
i2c.read(ADDRESS, *data, velikost dat, pokračování transakce);  
i2c.write(ADDRESS, *data, velikost dat, pokračování transakce);
```

Adresu je nutné posunout o 1 bit doleva (adresa << 1), neboť spodní bit je read/write indikace
Pokračování transakce slouží pro další zápis/čtení dat. 1 = pokračování, 0=ukončení

Sériová (UART) sběrnice

1. Definice Sériové sběrnice (globálně)

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

2. Nastavení přenosové rychlosti

```
uart.baud(rychlost);
```

3. Odeslání textu po sériové lince

```
uart.printf("text"); // použití standardní funkce printf  
uart.putc(znak); // odelsání jednoho znaku po sériové lince
```

4. Přijímání znaku po sériové lince

```
uart.getc();
```

Pro komunikaci po sériové lince je třeba využít některý z terminálů. Doporučuji používat putty, kdo chce, může i jiný, např. minicom, apod. Adresa samotného kitu je s největší pravděpodobností vždy **/dev/ttyACM0**. Spuštění terminálu je nutné s právy superuser (sudo putty). Abyste viděli, co do terminálu posíláte, doporučuji minimální formu echa v tomto tvaru (po definici sériové linky):

```
char c = uart.getc(); // načtení znaku z terminálu  
if(c != 0) { // přišel nějaký znak?  
    uart.putc(c);  
}
```

Analogově/Digitální převodník

1. Definice Analogového vstupu (globálně)

```
AnalogIn analog_in(vstup_pin); // AnalogIn analog_in(A0);
```

2. Vyčtení hodnoty z ADC a převod na napětí

```
float V;  
V = analog_in.read() * 3300;
```

PWM výstup

1. Definice PWM výstupu (globálně)

```
PwmOut pwm(pin); // PwmOut pwm(D5); - green RGB LED
```

2. Nastavení PWM periody a plnění (duty cycle)

```
pwm.period_ms(cas_v_ms); // Nastavení periody PWM v ms  
pwm.pulsewidth_ms(cas_v_ms); // Nastavení šířky pulzu v ms
```

ASCII znaky pro displej:

```
const unsigned char seven_seg_digits_decode_gfedcba[75]= {  
/* 0 1 2 3 4 5 6 7 8 9 : ; */  
0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F, 0x00, 0x00,  
/* < = > ? @ A B C D E F G */  
0x00, 0x48, 0x00, 0x00, 0x00, 0x77, 0x7C, 0x39, 0x5E, 0x79, 0x71, 0x7D,  
/* H I J K L M N O P Q R S */  
0x76, 0x30, 0x1E, 0x00, 0x38, 0x00, 0x54, 0x3F, 0x73, 0x00, 0x50, 0x6D,  
/* T U V W X Y Z [ \ ] ^ _ */  
0x78, 0x1C, 0x3E, 0x00, 0x00, 0x00, 0x5B, 0x00, 0x00, 0x00, 0x00, 0x08,  
/* ` a b c d e f g h i j k */  
0x00, 0x77, 0x7C, 0x58, 0x5E, 0x79, 0x71, 0x6F, 0x74, 0x30, 0x1E, 0x00,  
/* l m n o p q r s t u v w */  
0x30, 0x00, 0x54, 0x5C, 0x73, 0x67, 0x50, 0x6D, 0x78, 0x1C, 0x3E, 0x00,  
/* x y z */  
0x00, 0x6E, 0x5B  
};
```

Převzato z https://en.wikichip.org/wiki/seven-segment_display/representing_letters a poté upraveno pár písmen. Některé mají malý a velký ekvivalent.

Desetinná tečka má znak 0x80 a přidává se navíc ke znakům

Časovač

1. Definice časovače (globálně)

```
Ticker casovac; // Definice časovače
```

2. Nastavení časovače a jeho periody

```
// funkce zajišťující obsluhu časovače po jeho přetečení  
void callBack(void) {  
    // zde je výkonné jádro časovače  
}  
casovac.attach(&callBack, cas_v_s); // Nastavení periody PWM v ms
```