

PB171 – Semestral task description (spring 2021)

The semestral task is the only mandatory requirement to pass this subject. The task consists of two parts. The first part contains several schematics of the hardware you have to design. The second part is about implementation of API (application programming interface) for the selected peripherals.

1 General information

- The task will be submitted as one ZIP archive to the corresponding folder at IS MU. If you use a repository, you can simply compress and upload the whole repository. The link to the folder at IS MU:

<https://is.muni.cz/auth/el/fi/jaro2021/PB171/ode/111395642/>

- I recommend to use a repository during the semester and share the temporary results with me. We can avoid fatal mistakes and I have a feedback from you during the whole course.
- If I find any fatal mistake in the final submission, I will return it to you for a correction.
- After successful submission I will ask you for a defense as an online meeting. This defense must be completed before the end of the examination period. You will receive your mark immediately after your defense.

2 Hardware part

Please submit your electrical schematics as a PDF document. The PDF should contain images of all your schematics and every image should be commented about what is inside each schematics and why.

2.1 Abbreviations

- **CPU** – Your selected microcontroller approved by me. If you work with the PCB from the faculty, you should use PIC18F44K22.
- **LED** – Light-Emitting Diode
- **IO pin** – Input/output pin
- **GPIO pin** – General purpose input/output pin
- **AD converter** – Analog to digital converter
- **DA converter** – Digital to analog converter
- **RGB LED** – LED with red, green and blue channel
- **PWM** – Pulse Width Modulation
- **UART** – Universal asynchronous receiver-transmitter (specific peripheral of the CPU)
- **I2C** – Inter-Integrated Circuit or Two Wire Interface (specific peripheral of the CPU)
- **SPI** – Serial Peripheral Interface (specific peripheral of the CPU)
- **3S Li-Ion battery** – Li-Ion battery consists of three cells connected serially

2.2 Basic schematics

1. Draw schematics of the circuit with one LED driven by the CPU. The power supply is 5 V. Please specify a link to the datasheet of the used LED and the used CPU.
2. Draw schematics of the circuit with one switch and one button connected to the CPU as two input devices connected to two separate pins.
3. Draw schematics of the circuit with one button and one LED connected to any IO pin on the CPU. If the IO pin is set as output, the LED is driven by the CPU. If the IO pin is set as input, the CPU is able to read the state of the button and the LED indicates the state of the button, too.
4. Draw full schematics of the circuit with one LED driven by the CPU, one button connected as an input device to the CPU, power supply within the range 7 V-12 V, and all the passive components required by the CPU manufacturer.
5. Draw schematics of the circuit with one potentiometer connected as an analog input to the CPU. The CPU usually has several AD converters, you can use any of them.
6. Draw schematics of a thermistor connected to any AD converter on the CPU. The CPU should be able to measure the temperature of the thermistor. Please specify a link to the selected thermistor.
7. Draw schematics of the circuit with power input 7 V-12 V and one CPU. The CPU is able to measure the input voltage on its AD converter input. Please note that the CPU runs at lower voltage than the power supply and the AD converter cannot handle 12V directly. Please specify a link to the datasheet of every used component.
8. Draw schematics of the circuit with one RGB LED driven by the CPU. The CPU is able to set the brightness of every channel by PWM.

2.3 Schematics with high power consumption

9. Imagine that you just bought this servomotor:

https://hobbyking.com/en_us/turnigytm-tgy-50090m-analog-servo-mg-1-6kg-0-08sec-9g.html

You want to test the communication between the servo and your CPU. Draw a connection circuit between the servo and the CPU.

10. Imagine that we are building a small walking robot with 6 legs. Each leg needs 3 servos. You can find an example in this video:

<https://youtu.be/u115TyOnoFo>

Draw full schematics of the circuit with one 3S Li-Ion battery (or two batteries if needed), one CPU and 18 servos. You do not have to draw all the 18 servos, the most important part is how you solve the power supply.

11. Imagine that you just bought this DC motor:

<https://www.tme.eu/cz/en/details/sf-rob-11696/dc-motors/sparkfun-electronics-inc/rob-11696/>

You want to control it by CPU using PWM. Draw schematics of a connection between the CPU and the motor.

12. Change the previous schematics to allow to control the motor rotation in both directions.

13. Imagine that you have this motor:

https://hobbyking.com/en_us/turnigy-aerodrive-sk3-2826-980kv-brushless-outrunner-motor-1.html

with this speed controller:

https://hobbyking.com/en_us/blheli-s-20a.html

Draw schematics of the connection between your CPU used as the PWM source for this speed controller connected to this motor. Include the battery and provide a link to a suitable one.

2.4 Schematics of the peripherals

14. Draw schematics of this FTDI chip:

http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf

connected to the PC via USB and to your CPU via UART.

15. Imagine that you have this absolute orientation sensor:

https://cz.mouser.com/datasheet/2/783/BST_BMX055_DS000-1509552.pdf

Draw schematics of a connection between this sensor and your CPU using SPI interface.

16. Change the previous schematics to I2C interface.

17. Imagine that you have two CPUs. The distance between them is 20 meters. Draw schematics of the UART connection between them.

18. Draw schematics of a connection between an SD card and your CPU. You can select between SPI or SDIO interface.

3 Software part

The software API should be submitted as a full project with demo `setup()` and `loop()` functions. I have to be able to compile your solution and upload it to the real microcontroller.

If you use the PCB from the faculty or if you use the remote connection to the server, you can find some basic information including a template for your work here:

<https://github.com/bsaid/PB171>

You will implement the selected part of the Arduino API for your CPU as shown in the example here:

<https://github.com/bsaid/PB171/tree/main/ArduinoExamplePIC.X>

Please implement the functions below. Their documentation is available here:

<https://www.arduino.cc/reference/en/>

3.1 Platform independent functions

- `bit()`
- `bitClear()`
- `bitRead()`
- `bitSet()`
- `bitWrite()`

- `highByte()`
- `lowByte()`

3.2 Basic functions

- `pinMode()`
- `digitalWrite()`
- `digitalRead()`
- `analogRead()`
- `analogWrite()`
- `delay()`
- `delayMicroseconds()`
- `millis()`
- `micros()`

3.3 Class for communication with UART interface

Please follow the documentation here:

<https://www.arduino.cc/reference/en/language/functions/communication/serial/>

- `available()`
- `availableForWrite()`
- `begin()`
- `end()`
- `flush()`
- `parseInt()`
- `peek()`
- `print()`
- `println()`
- `read()`
- `readBytes()`
- `write()`

3.4 Advanced peripherals

Please select one of the following classes (ideally everybody different one) and implement it:

- <https://www.arduino.cc/en/Reference/EEPROM>
- <https://www.arduino.cc/reference/en/libraries/servo/>
- Design your own API to control the onboard display. You can find your inspiration, for example, here: <https://www.arduino.cc/en/Reference/LiquidCrystal>
- <https://playground.arduino.cc/Code/StopWatchClass/>