M2OS in Arduino Uno User's Guide

1. Run an example application on the board

Open the GNAT IDE (GPS) with the project for basic examples that use the M2OS API directly (they do not use the code transformation tool).

```
$ gps -P examples/api_m2os/examples_api_m2os_arduino_uno.gpr &
```

In order to launch the example in an Arduino Uno board plug the USB wire in the computer and press the button labeled "Run on board" (a long press will allow you to choose the example to build and launch).

The chosen application will be built and launched on the board. The serial output is shown in a window at the bottom part of the IDE.

The application is loaded in the board using `avrdude`, you may need to install it executing `sudo apt install avrdude`.

Although launching the application from inside the IDE is more convenient, you can also do it manually:

```
Send the application to the Arduino board:
$ avrdude -F -V -c arduino -p ATMEGA328P -P /dev/ttyACM0 -b 115200 -U flash:w:periodic_task.hex
In order to read the serial output of your program run:
$ stty -F /dev/ttyACM0 cs8 19200 ignbrk -brkint -icrnl -imaxbel -opost -onlcr -isig -icanon \
       -iexten -echo -echoe -echok -echoctl -echoke noflsh -ixon -crtscts
$ cat /dev/ttyACM0
```

Remote debugging is not available in the Arduino Uno board.

2. Test/debug the example on an emulator

This step requires you had installed the emulator `simulavr` (`sudo apt install simulavr`).

Press the button labeled "Run on simulavr" (a long press will allow you to choose the example to build and execute). The chossen application will be built and launched in the emulator. The emulator output is shown in a window at the botton part of the IDE.

Although running it from inside the IDE is more convenient, you can also execute manually the emulator using the command:

```
$ simulavr -d atmega328 -W 0xc6,- -T m2__end_execution__end_execution -f periodic_task
```

In order to debug the application running in the emulator press the button labeled "Debug on simulavr" (a long press will allow you to choose the example to debug).

Although debugging from inside the IDE is more convenient, you can also do it manually:

```
Run in a shell:
$ simulavr -d atmega328 -W 0xc6,- -T m2__end_execution__end_execution -f two_tasks_entry -g

Run GDB from other shell or from GPS:
$ avr-gdb
(gdb) tar rem :1212
(gdb) file two_tasks_entry
```

3. Using Arduino Uno libraries

There are plenty of free libraries for Arduino Uno available on Internet. In order to use any of those libraries the following steps must be performed

- Install the Arduino IDE.
- Install the library in the Arduino IDE as explained in the Arduino IDE documentation.
- Add the main header file of the library in variable LIBS_ALL of M2OS/arch/arduino_uno/drivers/libcore/Makefile
- Create an Ada binding for the functions of the library you plan to use in M2OS/arch/arduino_uno/drivers/libcore/ (in that directory there are examples you can look at)
- Edit tests/api_m2/arduino_uno/arduino_types_size_test.adb to check the size of type created in the binding ("with" the binding package and add one element to Arduino_CPP_Types and to Types_Data).
- Build the Ardunino core library executing:

```
M2OS/arch/arduino_uno/drivers/libcore/$ make
```

Bindings to some Arduino standard libraries, third party libraries and M2OS specific libraries are already provided:

- Control RC (hobby) servo motors (Servo Arduino standard library).
- Communicate with I2C / TWI devices (Wire Arduino standard library).
- Temperature and humidity sensor (DHT11.h).
- Zowi biped robot (Zowi.h).
- PIR Sensor.
- Buzzer (to play sounds and melodies) .
- CMPS03 I2C compass.
- HC-SR04 Ultrasonic Distance Sensor.
- EVShield to connect Lego EV3 and NXT sensors and motors to Arduino Uno.
- . . .

Arudino Uno examples can be found in examples/api_m2os/arduino_uno/.

You can check and run the examples by executing:

```
$ gps -P examples/api_m2os/arduino_uno/examples_api_m2os_arduino_uno.gpr
```

Zowi

Ada binding to the libraries for the BQ Zowi robot (https://www.bq.com/en/zowi)

Zowi examples can be found in examples/api_m2os/arduino_uno/zowi.

You can check and run the examples by executing:

```
$ gps -P examples/api_m2os/arduino_uno/examples_api_m2os_arduino_uno.gpr
```

4. Using AVR-Ada

It is possible to use AVR-Ada along with M2OS.

NOTE: Do not use AVR.Real_Time package. Use Ada.Real_Time instead (Rationale: AVR.Real_Time installs a timer handler, actually it is done in the separate unit AVR.Real_Time.Clock_Impl, which conflicts with the handler installed by M2OS)