Introduction to Embedded Systems (IES)

Module 2
Microcontroller programming

Version 2022-12-05, Kjeld Jensen kjen@sdu.dk

This module focuses on exploring the Arduino programming language and testing Arduino programming examples on the Raspberry Pi Pico microcontroller. You will learn about program structure, functions, variables, loops, control statements, serial communication etc.

Agenda

1. Review of module 1
2. Feedback on submitted reports
3. Microcontroller programming exercises
   A) Arduino program structure
   B) Functions
   C) Variables
   D) Loops
   E) Control statements and conditions
   F) Serial communication
   G) Arduino examples

A) Arduino program structure

The Arduino programming language is a subset of C/C++, with additional functionalities related to the microcontroller hardware.

An Arduino program is also called an *Arduino sketch*.

The Arduino sketch file must be named `.ino` and must be in a folder with the same name as the file name: if you have a sketch file named `hello_world.ino` it must be in a folder named `hello_world`

Program lines are terminated using a semicolon `;`

In a program line any text written after // is ignored by the Arduino compiler. This is often used for comments.

Question A.1) Please explain what is the difference between `setup()` and `loop()` in an Arduino program?

```
void setup() {
}

void loop() {
}
```

B) Functions

A function is a block of source code that only runs when it is called.

In Arduino programs the `setup()` and `loop()` are functions.

```
void setup() {
  // Setup the LED port
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  // Blink the LED once
  digitalWrite(LED_BUILTIN, HIGH);
  delay(500);
  digitalWrite(LED_BUILTIN, LOW);
  delay(500);
}
```

Above this text is the `hello_world` program from module 1.

Exercise B.1) Please modify `hello_world` to use functions for turning on and off the LED like in this example:

```
void setup() {
  // Setup the LED port
  pinMode(LED_BUILTIN, OUTPUT);
}

void turn_led_on() {
  // this function turns the LED on
  digitalWrite(LED_BUILTIN, HIGH);
}

void turn_led_off() {
  // this function turns the LED off
  digitalWrite(LED_BUILTIN, LOW);
}

void loop() {
  // Blink the LED once
  turn_led_on();
  delay(500);
  turn_led_off();
  delay(500);
}
```

Exercise B.2) Please modify `hello_world` to use a new function `wait_time()` to run the `delay` command. Your `loop()` function should then look like this example:

```
void loop() {
  // Blink the LED once
  turn_led_on();
  wait_time();
  turn_led_off();
  wait_time();
}
```

C) Variables

A variable is a container for storing data values such as a number or a text string. The variable content can be changed while the program is running.

Below is a list of common variable types. You will see that there are several different types for storing numbers depending on the minimum and maximum values and if it contains decimal numbers or not.

Type	Content
`char`	-127 to 128
`unsigned char`	0 to 255
`byte`	Same as `unsigned char`
`int`	-32,768 to 32,767
`unsigned int`	0 to 65,535
`long`	-2,147,483,648 to 2,147,483,647
`unsigned long`	0 to 4,294,967,295
`float`	Decimal number
`double`	Decimal number
`bool`	`true` / `false`
`boolean`	Same as `bool`
`String`	Text

Variables can be *global* or *local*. A global variable is available to the entire program. A local variable is only available inside a function.

A variable must be declared before it can be used.

Exercise C.1) Please modify `hello_world` to use a variable for the LED delay like in this example:

```
// declare the variable led_delay
int led_delay = 500;

void setup() {
  // Setup the LED port
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  // Blink the LED once
  digitalWrite(LED_BUILTIN, HIGH);
  delay(led_delay);
  digitalWrite(LED_BUILTIN, LOW);
  delay(led_delay);
}
```

D) Loops

A loop is used to execute a group of instructions or a block of code multiple times.

An example is the `for` loop:

```
for (statement1; statement2; statement3) {
}
```

`statement1` is executed before looping
`statement2` defines the condition for executing
`statement3` is executeed after the code block at each loop

Exercise D.1) Please modify `hello_world` to use a loop for blinking 10 times after startup like in this example:

```
void setup() {
  // Setup the LED port
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  // declare a local variable i
  int i;

  // Blink the LED 10 times
  for (i=0; i<10; i=i+1)
  {
    digitalWrite(LED_BUILTIN, HIGH);
    delay(500);
    digitalWrite(LED_BUILTIN, LOW);
    delay(500);
  }

  // wait 5 seconds
  delay (5000);
}
```

E) Control statements and conditions

A control statement is used to specify a block of code that is executed if a condition is met.

An example is the `if` statement:

```
if (condition) {
  // block that is executed if condition is true
}
```

Exercise E.1) Please modify `hello_world` to only blink when the BOOTSEL button is pressed like in this example:

```
void setup() {
  // Setup the LED port
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  // test if BOOTSEL button is pressed
  if (BOOTSEL == true)
  {
    // Blink the LED once
    digitalWrite(LED_BUILTIN, HIGH);
    delay(500);
    digitalWrite(LED_BUILTIN, LOW);
    delay(500);
  }
}
```

F) Serial communication

Serial communication between the Raspberry Pi Pico and your computer is a very efficient way to transmit data

Exercise F.1) Please create the program from the example below. Then modify the example to only send data to the serial port, when the BOOTSEL button is pressed.

```
void setup() {
  // Setup the serial device
  Serial.begin(115200);
}

void loop() {
  // Print text to the serial port
  Serial.println("Exercise F.1");
  delay (1000);
}
```

G) Arduino examples

The Arduino programming software includes many examples of programs.

You can find the list of examples under the `File -> Examples` menu. At the bottom of the list of examples you will find some examples that are specifically for the Raspberry Pi Pico microcontroller, but almost all of the built-in examples will work as well.

Exercise G.1) Please test some of the examples, and please where applicable use the knowledge from this module to modify the examples.