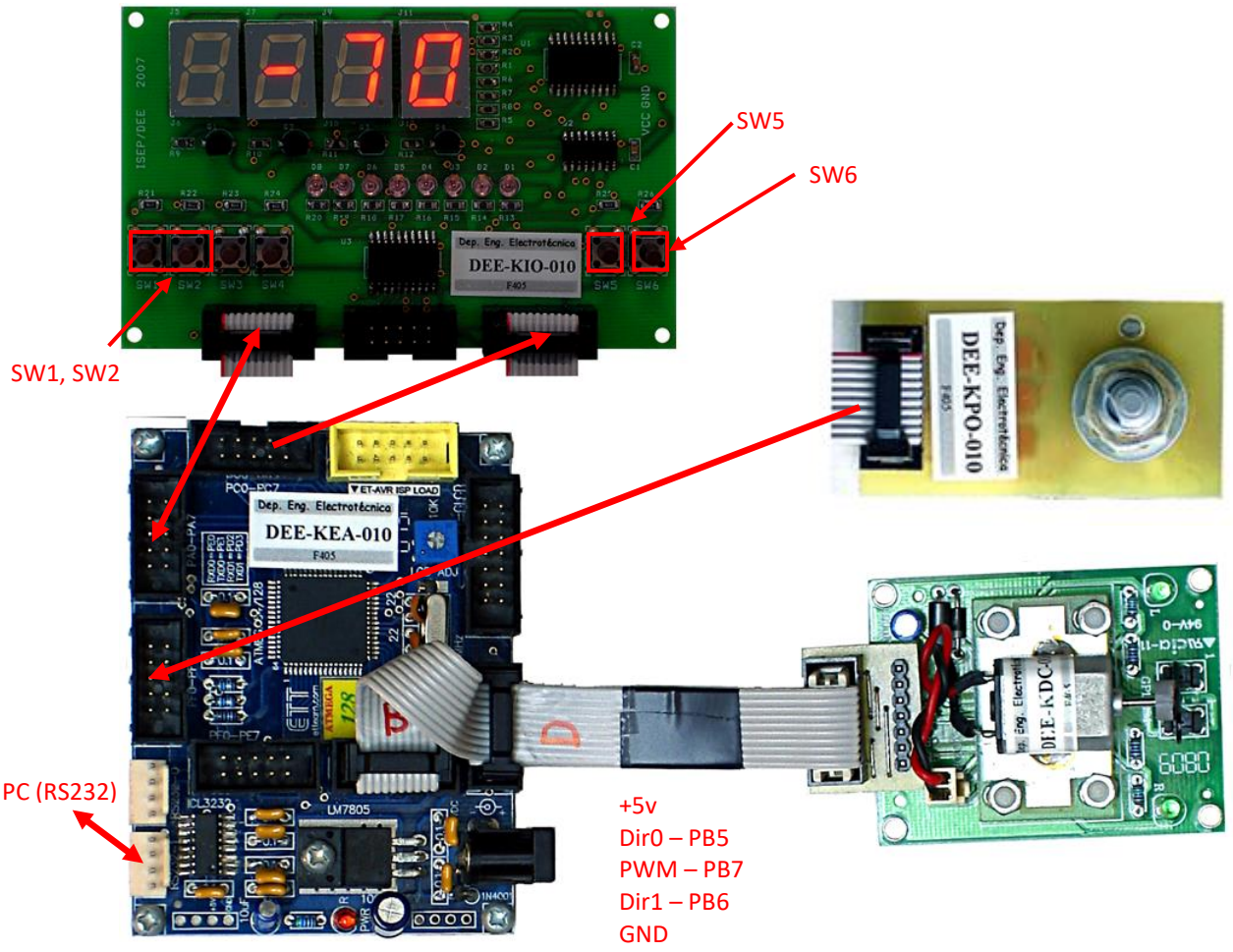
Project 2

Objective: Programming using C, open loop speed control of a DC motor, asynchronous serial communication, A/D converter and stepper motor control (MPP).

Requirements: Knowledge of C programming language, knowledge of: timer counters, interrupts, USART and A/D converter.

Operation 1: the objective is to control the speed of a DC motor using PWM. The switches **SW1, SW2,** increment (**SW1**) e decrement (**SW2**) by 5 points the percentage value of the nominal rotation speed of the motor. Switch **SW5** reverses the direction of the motor's rotation and **SW6** should stop the motor. Every time a switch is pressed, the motor's speed value (in percentage) should be updated in the display.

Necessary hardware:



Operation specifics:

The percentage value of the motors rotation speed is shown in the display. When the motor's rotation direction is counter clockwise, a sign (-) should be shown in *display* 2. To reverse the motor's rotation direction, the motor should be previously stopped for a period of **500 ms**.

Suggestion:

- Use the **TC0**, in **CTC** mode, to obtain a time base of **5 ms**.

- Use the **TC2, in PWM** mode (**Fast PWM or Phase Correct PWM**), to generate the PWM signal in the **OC2** output (PB7). The frequency of the PWM signal should be, approximately, **500 Hz**. The rotation direction is defined by the outputs **Dir0** (PB5) and **Dir1** (PB6).

Software implementation:

Use **C programming language.**

Operation 2:

Using ***USART1*** for the asynchronous serial communication (RS232), with **19200** bps, **8** bits of data and **1** stop bit, the objective is to control the motor's speed using the PC. The characters to be sent to the ATmega 128 are:

Character	Description
"P" or "p"	Stops the motor
"+"	Increments 5% of the nominal speed
"-"	Decrements 5% of the nominal speed
"I" or "i"	Reverses the rotation direction
"B" or "b"	Request to send the current duty cycle of the motor (in percentage) and the rotation direction

To select the mode of operation of the system, two new characters need to be added. The character "S" selects the *Switches* operating mode (controlled by the switches) and the character "D" selects the Digital operating mode (controlled by the PC).
The system must start with the Digital operating mode selected. The display 3 should indicate the current operating mode at all times.

Software implementation:

Use **C programming language.**

Operation 3:

The objective is to control the motor speed using the voltage given by a potentiometer, so that the variation of **0 V** to **Vcc** obtained at the potentiometer terminals should generate a a speed variation of 0 RPM to the nominal speed (0 V must correspond to 0% of the nominal speed and Vcc to 99% of the nominal speed). In this operating mode, the switch **SW5** reverses the rotation direction of the motor.

A new operation mode is added to the shown in display 3 (Analog Operating Mode – "A").

Character	Description
"S" or "s"	Switches operating mode
"D" or "d"	Digital operating mode
"A" or "a"	Analog operating mode

Analog data acquisition:

The analog/digital conversion of the voltage given by the potentiometer should be obtained by calculating the average value of 2 consecutive readings of the A/D converter using an accuracy of **8 bits**.
The data acquisition routine should be programmed in Assembly.

Software implementation:

Use **C programming language** + **Assembly programming language** (data acquisition routine)

Operation 4:

Using ***USART1*** for the asynchronous serial communication (RS232), with **19200** bps, **8** bits of data and **1** stop bit, the objective is to control the position of a stepper motor (20 steps/rotation or 40 half steps/rotation) using the commands sent by the PC. The character "**R**" commands the stepper motor to rotate to the right and the character "**L**" commands the stepper motor to rotate to the left. The motor's shaft can move from -180° to +180°. The character "**Z**" defines the position **0°** of the motor. The time between 2 consecutive steps must be approximately **25 ms**.
The characters to be sent to the ATmega 128 are:

Character	Description
"R"	Rotation of one step to the right
"L"	Rotation of one step to the left
"Z"	Definition of the position 0°
"0"	Rotate to the 0º position
"9"	Rotate to the 90º position
"1"	Rotate to the 180º position

Tables of the stepper motor drive modes:

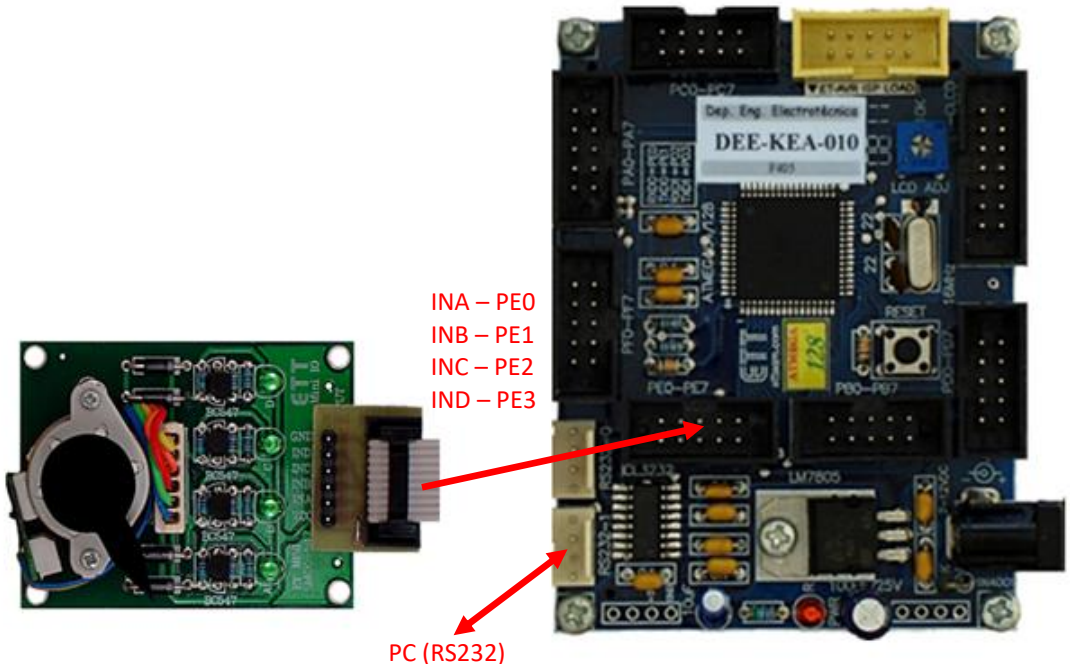
- Full step drive

IND	INC	INB	INA
1	0	0	1
1	1	0	0
0	1	1	0
0	0	1	1

- Half step drive

IND	INC	INB	INA
0	0	0	1
1	0	0	1
1	0	0	0
1	1	0	0
0	1	0	0
0	1	1	0
0	0	1	0
0	0	1	1

Necessary hardware:



Software implementation:

Use **C programming language.**