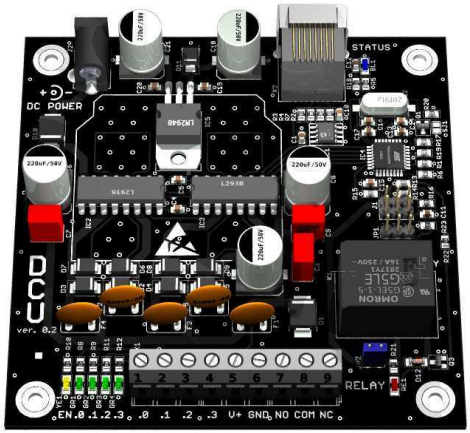
Digital Control Unit



Hardware description

Ver. 0.2

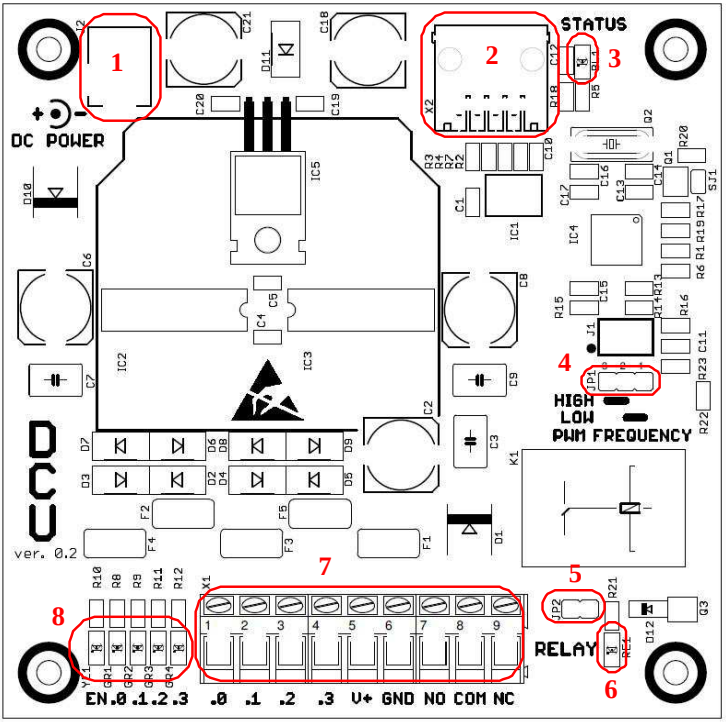
Electrical data:

Power supply (Vps):	6V– 15V DC /2A
Digital outputs max total current:	900mA
Digital output single line max current:	450mA
Digital outputs signal amplitude (load dependant):	≈ (Vps – 1V)
SPDT relay output:	30V AC / DC / 5A

Hardware functionality

- 4 independent push-pull digital outputs with PWM control and over-load protection
- 1 high current SPDT relay output controlled via OUTPUT ON / OFF signal
- All digital outputs and relay signals provided on screw terminal
- Vout screw terminal with over-current and over-voltage protection
- Protection against wrong power supply polarity
- High / low PWM frequency jumper switch
- Relay enable jumper switch
- 7 status control LED indicators
- Auto ID support
- Connection via RJ45 standard Edlab plug (ports J1…J4)

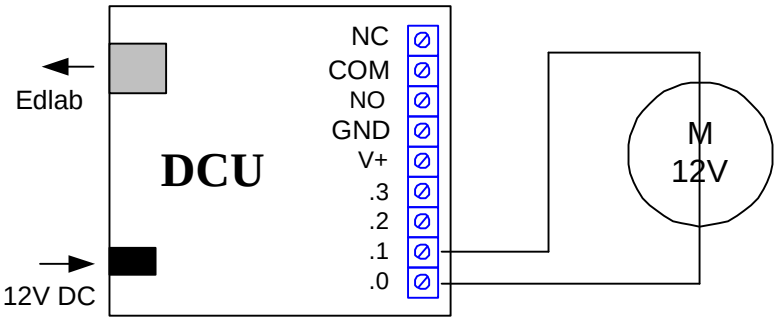
Board description:



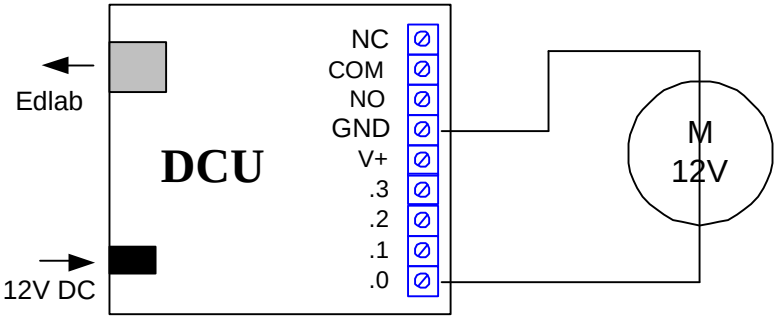
ELEMENT	ELEMENT DESCRITION
1	**Power supply socket.** Use 6-15VDC power adapter with min. 2A output current.
2	**Edlab connector.** Use RJ45 Straight through cable to connect the DCU board with the Edlab board. **Use ports J1..J4. Ports J5 and J6 don't support DCU board.**
3	**Status LED:** Slow blinking = waiting for inserting the DCU connection cable,

	Fast blinking = waiting for the “DCU OPEN” command, LED ON = “DCU OPEN” command received, DCU ready for operation
4	**PWM frequency jumper:** PWM frequency LOW = about 600Hz PWM frequency HIGH = about 5kHz
5	**Relay jumper:** Insert this jumper to control the relay via the “OUTPUT ON” / “OUTPUT OFF” command.
6	**Relay LED:** LED ON means relay ON, LED OFF means the relay is turned off
7	**Output terminal:** .0 – channel 0 output .1 – channel 1 output .2 – channel 2 output .3 – channel 3 output V+ – power supply output GND NO – relay output (normal open) COM – relay output (common terminal) NC – relay output (normal close)
8	**Signal LED indicators:** EN – output enabled indicator .0 – channel 0 status .1 – channel 1 status .2 – channel 2 status .3 – channel 3 status

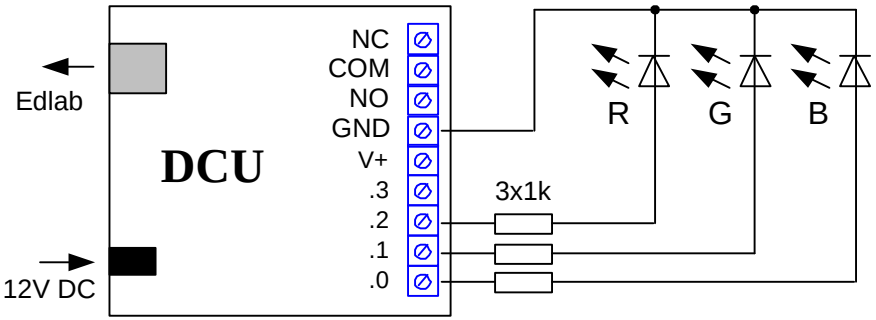
Wiring examples:



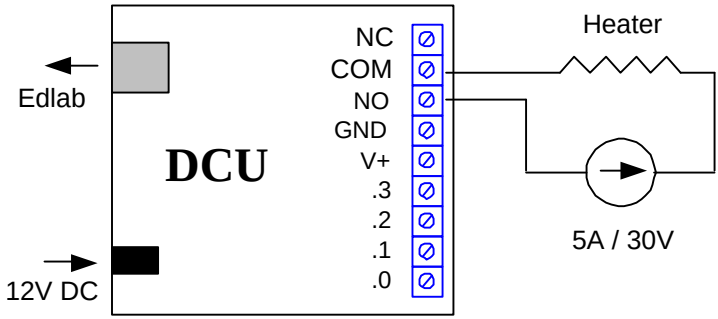
Example 1. DC motor control, variable speed and direction.



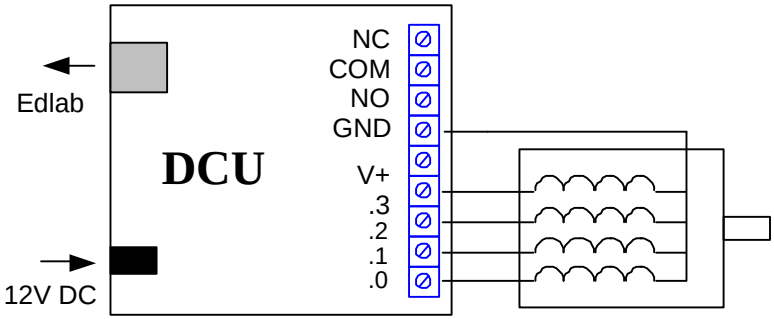
Example 2. DC motor control, variable speed.



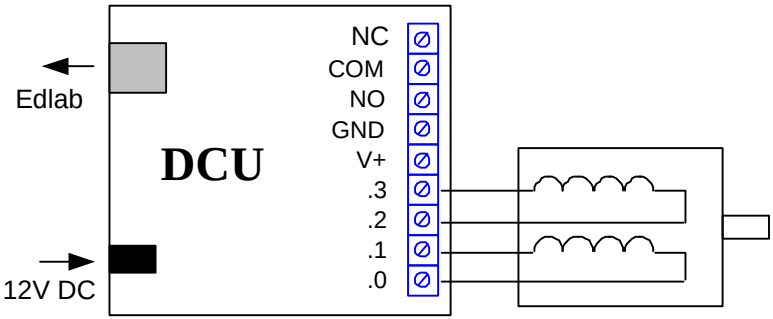
Example 3. RGB LED control.



Example 4. High current output usage.



Example 5. Unipolar stepper motor control.

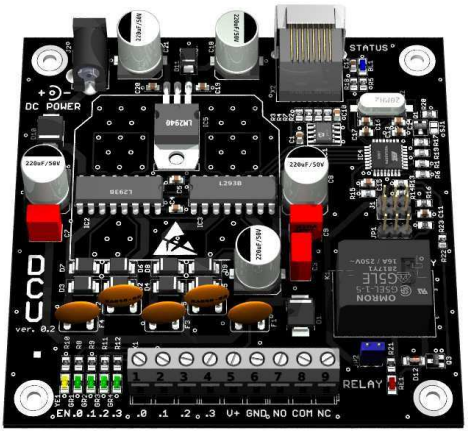


Example 6. Bipolar stepper motor control.

DIGITAL CONTROL UNIT

Software description

Ver 0.1



DCU script language fundamentals:

Reserved words (do not use them as labels, variables …) :

GP0, GP1, GP2 - Universal, general purpose sliders. The values of this sliders can be used to introduce some interaction between the program execution and the user via the user interface.

PB0, PB1, PB2, PB3 - Universal, general purpose push buttons. The binary states of this buttons can be used to introduce some interaction between the program execution and the user via the user interface.

J1, J2, J3, … J6 – Digital representation of the Edlab's analog sensors ports values. This values can be used to introduce some interaction between the program and the data received from the connected analog sensors.

JA, JB - Digital representation of the Edlab's digital sensors ports values. This values can be used to introduce some interaction between the program and the data received from the connected digital sensors.

The type and the range of the received values (as well from analog as from digital sensors) depends on the sensor type connected. While trying to pass the achieved value as a function parameter, don't forget to fit the range and the type. Use the INT or ABS function whenever it's

necessary. Simple division operation can help to get the proper maximal value.

T0 – The temperature measured via Edlab's internal high precision temperature sensor. This is the same value that is displayed on the bottom of the main form of Edlab's software.

Data types:

There is no specific data types supported by the DCU script language. You can use integer and float variables while working with simple arithmetic or relational operations. During passing a variable to the DCU output, make sure the type and the range fits the proper type and range you can use.

Data handling:

var = INT (source) – execution of this command will return the integer part of the source value. The fraction is the right side after the decimal point of a single. The integer is the left side before the decimal point. This function returns the integer part.

Example: value3 = INT(value2)
Value2 = 64,567 as a result the value3 = 64.

var = ABS (source) – execution of this command returns the absolute value of a numeric signed variable. The absolute value of a number is always positive.

Example: value3 = ABS(value2)
Value2 = -647 as a result the value3 = 647.

TOGGLE (source) – execution of this command inverts the state. The result is converted into binary data (0 or 1).

Example:
TOGGLE (source < 1) – the result is 1
TOGGLE (source >=1) – the result is 0

Variables declaration:

```
VAR value1                          //      value1
variable declaration
VAR temperature                     //
temperature variable declaration
```

Simple arithmetic:

```
Value3 = value1 + value2            // addition
Value3 = value1 - value2            // subtraction
Value3 = value1 * value2            //
multiplication
Value3 = value1 / value2            // division
```

Relational operators:

```
=                                   // equality
>                                   //      greater
than
```

```
<                                   // less than
<>                                  // inequality
```

Loops:

```
DO                                  // repeat a
block of statements
        IF VALUE1 > 10 THEN
                EXITDO              // exiting the
        structure
        ENDIF
        ……..
LOOP
```

Conditional executions:

```
IF VALUE1  >  10 THEN               // conditional
execution
        OUTPUT DEC(10)
ELSE
        OUTPUT DEC(0)
ENDIF
……

IF VALUE2  <  55 THEN               // conditional
execution
        OUTPUT BIN (1,0,0,1)
ENDIF
```

Labels:

GOTO JP1 // jump to the specified label

LABEL JP1:

Linking with sensors:

DO
Value2 = J2 / 50 // range scaling via division
Value2 = ABS(Value2) // absolute value conversion
OUTPUT PWM (0,Value2) // channel 0 controlled as a PWM output
LOOP

Delays:

WAIT (VAL) – execution of this command will suspend the execution of the script for a value defined via the VAL parameter.

Example: WAIT (1-100)
WAIT (10) will force the interpreter to sleep for the time of 10 DCU STEPs.

DCU dedicated commands:

DCU STEP (VAL) – this command precise the speed of the execution of the DCU scripts. Increasing the value will slower the execution of the script. Place this command in the beginning of each program. The DCU

STEP slider overrides the value précised via the command placed in the script.

Example: DCU STEEP (100-1000) – value given in ms
DCU STEP (100) means that the script is executed in 100ms / command tact. Every command takes 100ms.

DCU OPEN – this command opens the communication channel between Edlab and the DCU board. Without sending this command the DCU board will ignore all the received data. This command should be placed in the beginning of each program. This command must be sent after each DCU POWER ON.

DCU RESET – this command causes the reset of the DCU board. All outputs will be turned off.

OUTPUT ON – execution of this command will turn the DCU outputs ON. The enable and relay LEDs will be turned ON, If the relay jumper is inserted, the relay will be also turned on.

OUTPUT OFF – execution of this command will turn the DCU outputs OFF. The enable LED, the relay / relay LEDs will be turned OFF.

OUTPUT PWM (CH,VAL) – execution of this command will turn on the PWM generation. A square wave will be generated on channel CH, the duty cycle will be VAL %.

Example: OUTPUT PWM (0…3, 0…100)
OUTPUT PWM (0,50) means that the channel .0 will generate a square wave with the frequency selected by the PWM frequency jumper, the duty cycle will be 50%.

OUTPUT BIN (CH0, CH1, CH2, CH3) – execution of this command will set the state of the DCU outputs. Each output channel will be determined by a binary value.

Example: OUTPUT BIN (0/1, 0/1, 0/1, 0/1)
OUTPUT BIN (0,0,1,1) means that there will be no voltage (about 0V) on channel .0 and .1. The voltage on channels .2 and .3 will approximately reach the value of the power supply voltage.

OUTPUT DEC (VAL) – execution of this command will set the state of the DCU outputs. Each output channel will be determined by a binary value coded via a decimal number from 0 to 15.

Example: OUTPUT DEC (0…15)
OUTPUT DEC (12) means that there will be no voltage (about 0V) on channel .0 and .1. The voltage on channels .2 and .3 will approximately reach the value of the power supply voltage.

OUTPUT DEC (12) = OUTPUT BIN (0,0,1,1)

$12 = 0 * 2^0 + 0 * 2^1 + 1 * 2^2 + 1 * 2^3$

Auxiliary commands:

GP0, GP1, GP2 MIN - this command precise the minimal value that can be set with the GP… slider.

GP0, GP1, GP2 MAX - this command precise the maximal value that can be set with the GP… slider.

DCU script language program examples:

PROGRAM NAME	PROGRAM DESCRIPTION
PB.dcu	Push buttons control digital outputs.
SLD1.dcu	Slider controls digital outputs.
SLD2.dcu	Slider controls PWM outputs.
SLD_RGB.dcu	RGB LED controller.
TC.dcu	Simple temperature controller.

DC_MOTOR.dcu	DC motor control sample program.
STEPPER1.dcu	Unipolar stepper motor sample program.
XMAS.dcu	Simple xmas light demo.
SENSORS.dcu	Redirecting data from sensors to PWM outputs
PEAK_DET.dcu	Peak detector sample program.

All the sample programs can be found here: EdLaB -> DCU Samples.

Connexia electric, s.r.o.
735 71, Dětmarovice 1170

www.connexia.cz
edlab@connexia.cz