

Přílohy

Příloha 1: Parametry 12V DC motoru

1. POPIS

Tento stejnosměrný motor vyznačuje především dobrým točivým momentem. Zařízení lze využít např. v Arduino projektech jako druh pohonu. Motor je osazen kovovou hřídelí, která není kulatého tvaru (dvě ploché hrany). Pro snadné zakomponování do aplikací zařízení disponuje montážními otvory.

Základní charakteristika:

- Napájení 12 V DC
- Až 14 otáček za minutu
- Točivý moment 1,25 kg.cm
- Plastová převodovka (polyoxymetylen)
- Montážní otvory

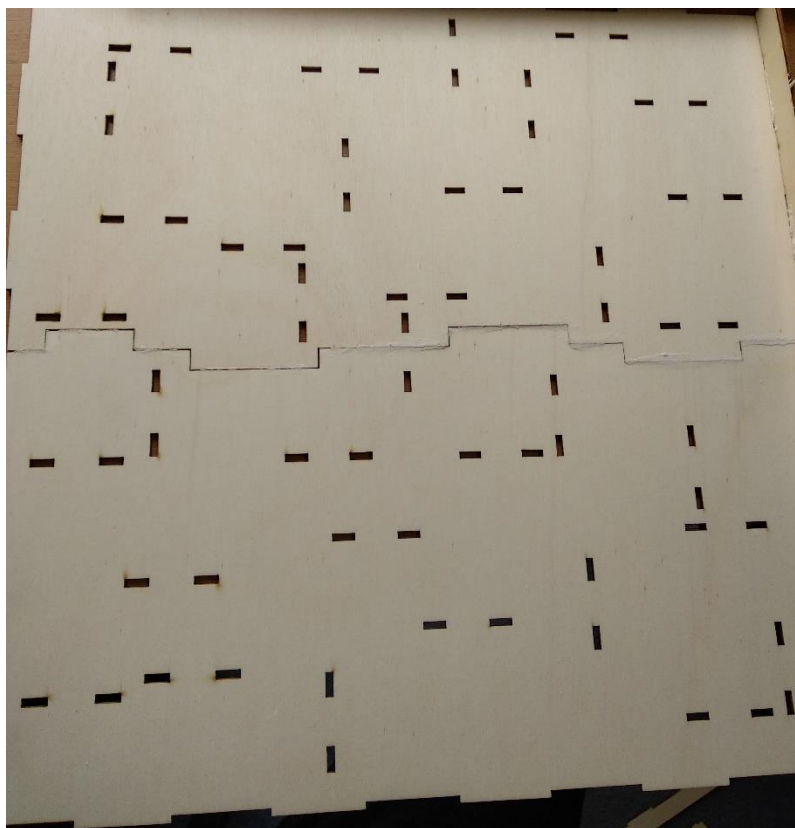


2. SPECIFIKACE

Model	S30K	Točivý moment	1,2 kg.cm
Napájení motoru	12 V DC	Rozměry hřídele (mm)	8 x 5 x 3,7
Proud naprázdno	cca 50 mA	Rozteč mont. otvorů (mm)	37,5
Proud v zátěži	cca 130 mA	Průměr mont. otvorů (mm)	3
Max. rychlost otáčení	14 RPM	Rozměry (mm)	34,5 x 32

Příloha 2: Komponenty pracovní desky

Pracovní deska



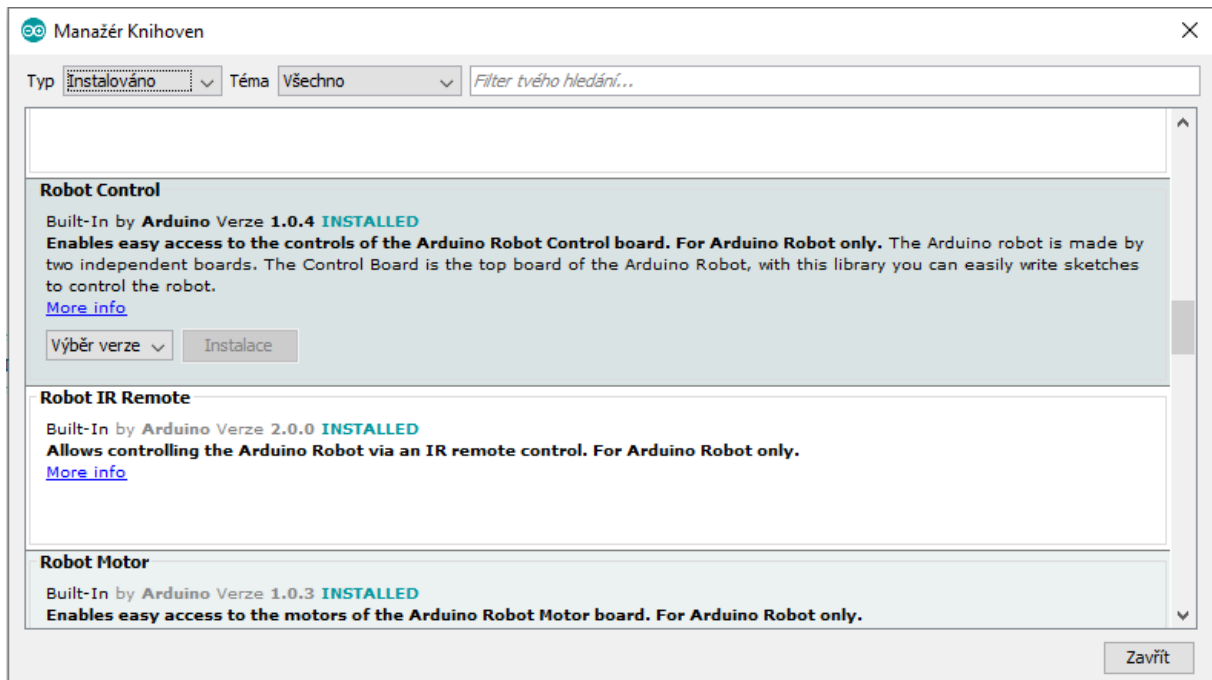
Boční strana



Zarážka



Příloha 3: Knihovna Arduino



Příloha 4: Ovládání motoru v Arduino

```
L298N_DC_Motor_control | Arduino 1.8.12 (Windows Store 1.8.33.0)
Soubor Upravy Projekt Nástroje nápověda

L298N_DC_Motor_control

const int IN1=8;
const int IN2=7;
const int EN1=5;
const int Potentiometer=A0;
const int IN3=4;
const int IN4=3;
const int EN2=6;
const int Potentiometer2=A1;
int MotorSpeed=0;
int MotorSpeed1=0;
char datas;
String Direction="";
void setup()
{
  Serial.begin(9600);
  pinMode(IN1,OUTPUT);
  pinMode(IN2,OUTPUT);
  pinMode(EN1,OUTPUT);
  pinMode(IN3,OUTPUT);
  pinMode(IN4,OUTPUT);
  pinMode(EN2,OUTPUT);
  Serial.println("*****");
  Serial.println("                L298N DRIVER SPEED AND DIRECTION CONTROL                ");
  Serial.println("*****");
  digitalWrite(IN3,LOW);
  digitalWrite(IN4,LOW);
}

void loop()
{
  while(Serial.available())
  {
    delay(10);
    datas=Serial.read();
    Direction+=datas;
  }
}
```

```
L298N_DC_Motor_control | Arduino 1.8.12 (Windows Store 1.8.33.0)
Soubor Upravy Projekt Nástroje nápověda

L298N_DC_Motor_control$

//----- SPEED CONTROLL -----
MotorSpeed=map(analogRead(Potentiometer), 0, 1023, 0, 255); // Read the speed from the potentiometer analog value from 0 to 5V ==> 0 to 255
analogWrite(EN1, MotorSpeed);
MotorSpeed1=map(analogRead(Potentiometer2), 0, 1023, 0, 255); // Read the speed from the potentiometer analog value from 0 to 5V ==> 0 to 255
analogWrite(EN2, MotorSpeed1); // Write the speed converted speed to PWM output

//----- DIRECTION CONTROLL -----
if(Direction=="down") // Left direction command
{
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);

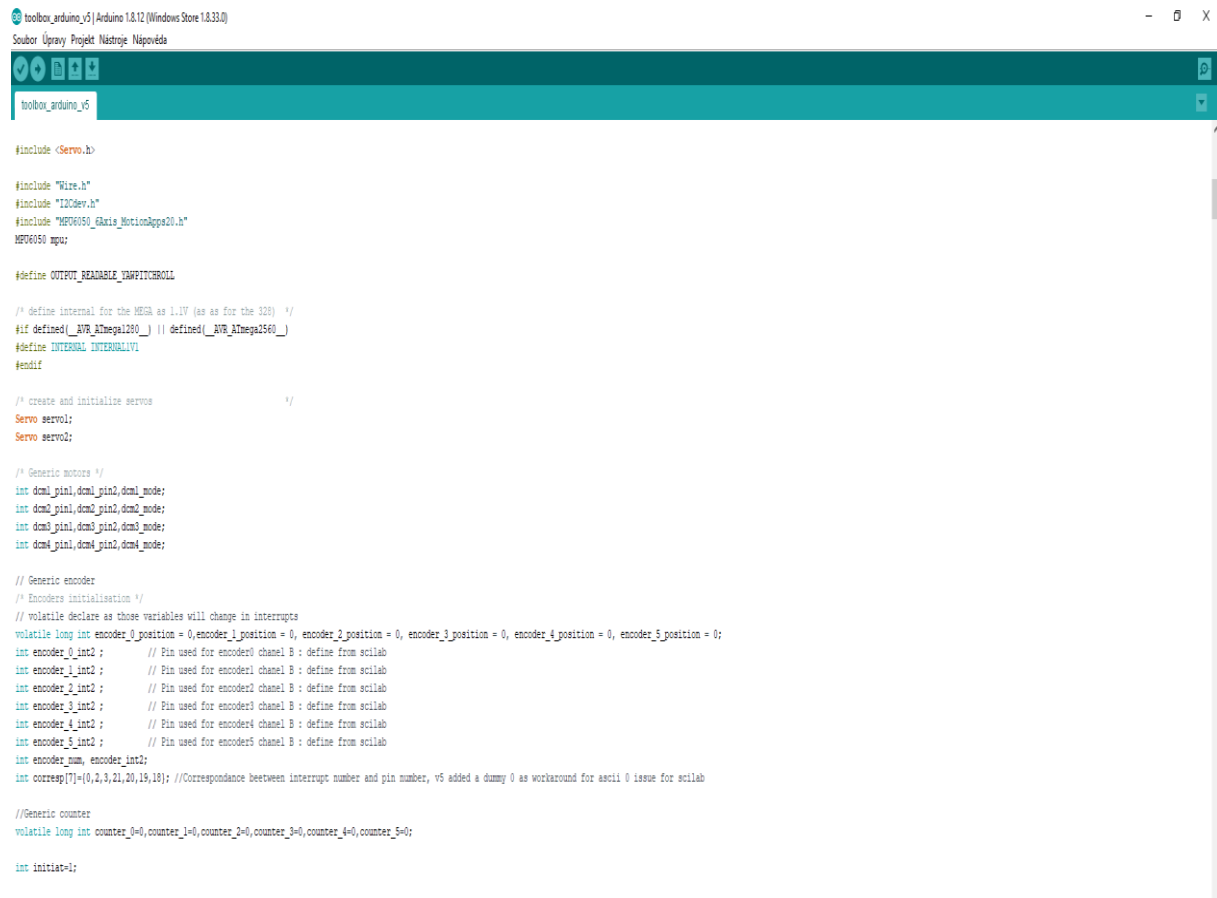
  Serial.println("down = Motor run left direction");
}
if(Direction=="up") // Right direction command
{
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);

  Serial.println("up = Motor run right direction");
}
if(Direction=="down1") // Left direction command
{
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  Serial.println("down1 = Motor1 run left direction");
}
if(Direction=="up1") // Right direction command
{
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
  Serial.println("up1 = Motor1 run right direction");
}

Direction="";
}
```

Příloha 5: Program v Arduino pro spárování s programem Scilab

Program je stažitelný ze stránky: <https://atoms.scilab.org/toolboxes/arduino>



```
#include <Servo.h>

#include "Wire.h"
#include "I2Cdev.h"
#include "MP06050_6Axis_MotionApps20.h"
MP06050 mpu;

#define OUTPUT_READABLE_YAWPITCHROLL

/* define interval for the MEG as 1.1V (as as for the 328) */
#if defined(_AVR_ATmega1280_) || defined(_AVR_ATmega1560_)
#define INTERVAL INTERVAL1V
#else
#define INTERVAL INTERVAL1V
#endif

/* create and initialize servos */
Servo servo1;
Servo servo2;

/* Generic motors */
int dcm1_pin1,dcm1_pin2,dcm1_mode;
int dcm2_pin1,dcm2_pin2,dcm2_mode;
int dcm3_pin1,dcm3_pin2,dcm3_mode;
int dcm4_pin1,dcm4_pin2,dcm4_mode;

// Generic encoder
/* Encoders initialization */
// volatile declare as those variables will change in interrupts
volatile long int encoder_0_position = 0, encoder_1_position = 0, encoder_2_position = 0, encoder_3_position = 0, encoder_4_position = 0, encoder_5_position = 0;
int encoder_0_int2 ; // Pin used for encoder0 channel B : define from scilab
int encoder_1_int2 ; // Pin used for encoder1 channel B : define from scilab
int encoder_2_int2 ; // Pin used for encoder2 channel B : define from scilab
int encoder_3_int2 ; // Pin used for encoder3 channel B : define from scilab
int encoder_4_int2 ; // Pin used for encoder4 channel B : define from scilab
int encoder_5_int2 ; // Pin used for encoder5 channel B : define from scilab
int encoder_num, encoder_int2;
int corresp[]={0,2,3,21,20,19,18}; //Correspondance between interrupt number and pin number, v5 added a dummy 0 as workaround for ascii 0 issue for scilab

//Generic counter
volatile long int counter_0=0,counter_1=0,counter_2=0,counter_3=0,counter_4=0,counter_5=0;

int initiat=1;
```



```

toolbox_arduino_v5 | Arduino 1.8.12 (Windows Store 1.8.33.0)
Soubor Upravy Projekt Nástroje nápověda

toolbox_arduino_v5
}
if (Serial.available() != 0) {
  val = Serial.read();
}

//while (Serial.available() == 0) {} // Waiting char
//val = Serial.read();

// if (val == 0) // version
// Serial.print("v3");
// val--;
// }
//case A -> Analog
if (val == 65) { // A -> Analog read
  while (Serial.available() == 0) {} // Waiting char
  val = Serial.read();
  if (val == 14) { // '1' -> read pin
    while (Serial.available() == 0) {} // Waiting char
    val = Serial.read();
    if (val > 47 && val < 67) { // from pin 0, to pin 19
      pin = val - 48; // number of the pin
      agv = analogRead(pin);
      //Serial.println(agv);
      //Serial.print(agv);
      Serial.write((uint8_t*) &agv, 2); /* send binary value via serial */
    }
    val--;
  }
}
else if (val == 67) { // W -> Analog write
  while (Serial.available() == 0) {} // Waiting char
  val = Serial.read();
  if (val > 47 && val < 67) { // from pin 0 to pin 19
    pin = val - 48; // number of the pin
    while (Serial.available() == 0) {} // Waiting char
    val = Serial.read();
    analogWrite(pin, val);
  }
  val--;
}
//}

//case D -> Digital
else if (val == 68) { // D -> Digital pins
  while (Serial.available() == 0) {} // Waiting char
  val = Serial.read();
}
}
}

```

Příloha 6: knihovna Arduino pro spárování programu Scilab s platformou Arduino

The screenshot shows the ATOMS package manager interface. On the left is a sidebar with a tree view of categories, including 'All modules', 'Bioinformatics', 'Data Analysis', 'GUI', 'Real-Time', and 'Xcos'. The 'Arduino' package is selected, and its details are shown in the main panel. The details include the version (1.8), author (Bruno JOFRET), and a description stating it provides Arduino communication blocks for Xcos. It also lists tutorials and resources, such as 'Blinking LED Tutorial' and 'Temperature monitoring Tutorial'. At the bottom, there are buttons for 'Autoload', 'Install', 'Remove', and 'Back'.

Main categories - ATOMS

File ?

- All modules
- Bioinformatics
- Contributed Scilab builds
- Data Analysis
- Data Handling
- Documentation
- Domain-Specific
- Education
- GUI
- Graphics
- Graphs
- Image Processing
- Instruments Control
- Linear algebra
- Manuels
- Modeling
- Number theory
- Numerical Maths
- Optimization
- Parallel Computing
- Physics
- Real-Time
- Scilab development
- Signal Processing
- Technical
- Windows Tools
- Xcos

Arduino

NOTE

* toolbox_arduino_v5.ino is shipped within the toolbox, including the MPU6050 bloc and filtering. Thanks to David VIOLEAU.

CHANGES 1.7.1 => 1.8

=====

BUGS FIXED

* Fixed issue of extra figure popup when demo figure is on

NEW FEATURES

* Add command lines control support for common interface.

* Translate help pages in english

* Translate demos

TODO:

* Change the firmware to get multiple channels and sent all back to host for performance improvement

* Translate blocks interfaces and error messages

CHANGES 1.7 => 1.7.1

=====

BUGS FIXED

* Fixed 2206 Problem with Servo_Write Block

* Restore MPU6050 block

CHANGES 1.6.2 => 1.7

=====

BUGS FIXED

* Fixed Scilab 6 compatibility issues

* Change the serial to Scilab from char to double to solved ascii issues.

* Fixed port >9 not supported issue

Autoload

Install Remove Back

Main categories - ATOMS

File ?

- All modules
- Bioinformatics
- Contributed Scilab builds
- Data Analysis
- Data Handling
- Documentation
- Domain-Specific
- Education
- GUI
- Graphics
- Graphs
- Image Processing
- Instruments Control
- Linear algebra
- Manuels
- Modeling
- Number theory
- Numerical Maths
- Optimization
- Parallel Computing
- Physics
- Real-Time
- Scilab development
- Signal Processing
- Technical
- Windows Tools
- Xcos

Arduino

* Translate demos

CHANGES 1.5 => 1.6.2

=====

BUGS FIXED

* toolbox_arduino_v4.ino file shipped in the module was bad. Replaced with the right file.

* ARDUINO_pre_simulate() was expecting arduino_v3.ino instead of arduino_v4.ino and yielded an error 'Il faut charger avec le logiciel arduino le sketch toolbox_arduino_v3.ino dans la carte Arduino' when actually dealing with the v.4

* The DESCRIPTION file that was removed from v.1.4 is actually needed to instal the module offline. Restored.

* Demos: Examples for Scilab 6.0 are displayed with wrong ports positions with Scilab 5.5. The versions for Arduino 1.1 are restored in the binary for the Scilab 5.5.

TODO:

* Extend the module to Linux users (through the serial toolbox based on TCL)

* Translate help pages in english

* Translate blocks interfaces and error messages

* Translate demos

See also

- <http://atoms.scilab.org/toolboxes/arduino/1.8>
- <http://www.scilab-enterprises.com>

Release date

2020-02-16

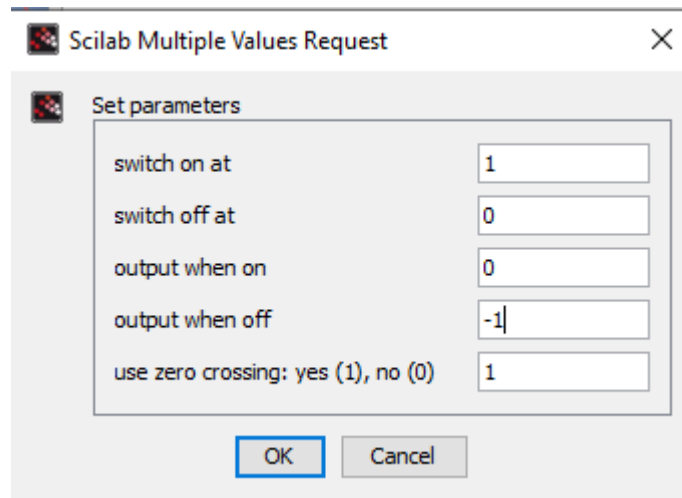
Download size

1.7 MB

Autoload

Install Remove Back

Příloha 7: Nastavení hystereze



Příloha 1: Parametry 12V DC motoru	1
Příloha 2: Komponenty pracovní desky	2
Příloha 3: Knihovna Arduino	3
Příloha 4: Ovládání motoru v Arduino	4
Příloha 5: Program v Arduino pro spárování s programem Scilab	5
Příloha 6: knihovna Arduino pro spárování programu Scilab s platformou Arduino	7
Příloha 7: Nastavení hystereze	9