Maxon Motor & Motor Controller Manual

Note: This manual is only for use for the Maxon motor and controller outlined below. This information is based upon the tutorial videos found online and through testing.

NOTE: Maximum Permitted Supply Voltage: 10 – 70 VDC. Do not exceed 76 VDC or wrong polarity, or it will destroy the unit. Minimum output voltage is 8 VDC. Continuous Max Output Current: 10 A / short-time (acceleration, < 20 seconds) max. 30 A.

Necessary Equipment:

- Motor (EC 90 flat Ø90 mm, brushless, 90 Watt, with Hall sensors, PN # 323772)
- Motor Controller (ESCON 70/10, 4-Q Servocontroller for DC/EC Motors, 10/30 A, 10 – 70 VDC, PN # 422969)
- Maxon Motor Controller Software (ESCON Setup – Studio 2.2/Firmware 0150h)
- Suitable Cables for Power (J1) : 0.2 – 2.5 mm² multi-core & single wire, AWG 24-12
- Suitable Cables for Motor (J2) : 0.2 – 2.5 mm² multi-core & single wire, AWG 24-12
- Suitable Cables for Hall Sensors (J3) : 0.14 – 1.5 mm² multi-core & single wire, AWG 28-14
- J4 will not be used. The HABIP motor does not have an encoder.
- Suitable Cables for Digital I/Os (J5) : 0.14 – 1.5 mm² multi-core & single wire, AWG 28-14
- Suitable Cables for Analog I/Os (J6) : 0.14 – 1.5 mm² multi-core & single wire, AWG 28-14

Software Set-Up (For Testing with Arduino Uno):

- Connect the servocontroller to the computer that will be running the motor software via USB.
- Open ESCON Studio 2.2.
- The Startup Wizard should begin automatically.
  - If this does not start automatically, go to the “Tools” dropdown menu and select “Setup Wizard.”
  - Read the safety instructions and select “Yes, I have read the above instructions.” Click “Next.”
  - A description of the device connected is shown. Double-check to ensure that it is the correct motor controller. Click “Next.”
  - Select “Maxon EC Motor.” Click “Next.”
  - Enter the motor data that is needed.
    - For the motor being used on the HABIP, the constants that need to be inputted are as follows:
- Speed Constant: 135 RPM/V
- Thermal Time Constant Winding: 34.1 s
- Number of Pole Pairs: 12
  - Click “Next.”

Enter the system data that is needed.
  - Note: This is not necessarily just based upon the motor. Ensure that the values inputted here are the maximums for the entire system so as to ensure that the system is not harmed.
  - The values for the HABIP systems that are to be inputted are as follows:
    - Max Permissible Speed: 5000 RPM
    - (3190 RPM No Load Speed) 2590 RPM
    - Nominal Current: 6.0000 A
    - Max Output Current Limit: 30.0000 A
  - Click “Next.”

The next page is entitled “Detection of Rotor Position.”
  - Under the “Select Type of Sensor” drop-down menu, click “Digital Hall Sensors.”
  - Select “maxon” for “Hall Sensor Polarity.”
  - Click “Next.”

Select the type of sensor being used for measuring the speed.
  - For the motor being utilized by the HABIP teams, select “Available Hall Sensors,” as there is no encoder.
  - Click “Next.”

Select the desired mode of operation.
  - There are three modes:
    - Current Controller
    - Speed Controller (Closed Loop)
    - Speed Controller (Open Loop)
  - Select “Speed Controller (Closed Loop)” for the HABIP motor.
  - Click “Next.”

This next page is the “Enable” page. This page is used to determine the input necessary to start up the motor.
  - There are several “Enable” functionalities:
    - Enable
    - Enable & Direction
    - Enable CW
    - Enable CCW
    - Enable CW & CCW
There is a section to assign this functionality:

- Select a digital input (1, 2, 3 or 4) to assign the “Enable” functionality to it.
  - For the HABIP motor, assign it to “Digital Input 2.”
- The polarity must also be specified as either “High Active” or “Low Active.”
  - For the HABIP motor, specify the polarity as “High Active.”
- Click “Next.”

This page will define the input functionality.

- There are several options for inputs:
  - Analog Set Value
  - PWM Set Value
  - 2 Fixed Set Value
  - Fixed Set Value
- For the HABIP motor, assign the input as PWM Set Value
- Speed at 10% : -5000.0 RPM
- Speed at 90% : 5000.0 RPM
- Click “Next.”

This next page is to set the current limit on the motor.

- There are several types of current limit functionality:
  - Analog Current Limit
  - PWM Current Limit
  - 2 Fixed Current Limit
  - Fixed Current Limit
- Specify the maximum value for current.
  - For HABIP, select “Fixed Current Limit.”
  - Write down “30.0000 A.”
- Click “Next.”

The speed ramp page allows for several possibilities for the motor:

- Analog Ramp
- Fixed Ramp
- No Ramp Active
  - Select “Fixed Ramp” and write 1000 RPM/s for both the acceleration and deceleration of the HABIP motor.
- Click “Next.”

The minimal speed page is used to allow for a low speed to be the

- Write “0.0 RPM” for the “Minimal Speed.”
For the offset page, there are multiple options:

- Analog Offset
- PWM Offset
- RC Servo Offset
- Fixed Offset
  - Select “Fixed Offset” and set to 0.0 RPM.
  - Click “Next.”

This next page will set up the digital inputs and outputs of the motor controller system.

- Note: One of the digital inputs will be marked as “Enable” if it has been selected earlier as one.
  - Digital Input 1:
    - None
    - Stop
  - Digital Input 2:
    - None
    - Stop
  - Digital I/O 3:
    - Stop
    - Ready
    - Speed Comparator
    - Current Comparator
  - Digital I/O 4:
    - Stop
    - Ready
    - Speed Comparator
    - Current Comparator

Set up the motor controller as follows for the HABIP motor:

- Digital Input 1: PWM - Set Value
- Digital Input 2: Enable
- Digital I/O 3: None
- Digital I/O 4: None

- Note: Be sure to double-check that the inputs are within the constraints set by Maxon. The digital inputs and digital inputs/outputs have different requirements.
- Click “Next.”

The analog inputs will be selected at this page:

- Analog Input 1: Actual Speed
● Analog Input 2: Actual Current Averaged
  ▪ Click “Next.”
  ○ There are three analog outputs available from the controller.
    ▪ They have multiple options for functionality:
      ● None
      ● Fixed Value
      ● Actual Current
      ● Actual Current Averaged
      ● Actual Speed
      ● Actual Speed Averaged
      ● Demand Current
      ● Demand Speed
      ● Temperature Power Stage
      ▪ Click “Next.”
  ○ This next page will be used to specify any analog outputs that were chosen in the previous step.
    ▪ Speed at: 0.000 V : -5000.0 RPM
    ▪ Speed at 3.000 V : 5000 RPM
    ▪ Click “Next.”
    ▪ Current at 0.000 V : 0.0000 A
    ▪ Current at 3.000 V : 10.0000 A
  ○ This next page shows the configuration summary.
    ▪ Click “Show Wiring Overview.”
      ▪ This will show the wiring overview for the module based upon the configuration determined by the options chosen above.
Wiring Diagram

**ESCON 70/10 – Wiring Overview**

On the following pages you will find the wiring information based on the configuration you performed in «ESCON Studio».

![Wiring diagram](image)

**Remark**
- Downward arrow: Ground safety earth connection (optional)

![Wiring diagram](image)

**Remark**
- Downward arrow: Ground safety earth connection (optional)

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
<th>Configured Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>−</td>
<td>Power_GND</td>
<td>Ground of supply voltage</td>
</tr>
<tr>
<td>+</td>
<td>+Vcc</td>
<td>Power supply voltage (+10…+70 VDC)</td>
</tr>
</tbody>
</table>
### J2 Motor

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
<th>Configured Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DC: Motor (+M)</td>
<td>EC motor: Winding 1</td>
</tr>
<tr>
<td></td>
<td>EC: Motor winding 1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>DC: Motor (-M)</td>
<td>EC motor: Winding 2</td>
</tr>
<tr>
<td></td>
<td>EC: Motor winding 2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>DC: not connected</td>
<td>EC motor: Winding 3</td>
</tr>
<tr>
<td></td>
<td>EC: Motor winding 3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Motor shield</td>
<td></td>
</tr>
</tbody>
</table>

### J3 Hall Sensor

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
<th>Configured Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Hall sensor 1</td>
<td>Hall sensor 1 input</td>
</tr>
<tr>
<td>2</td>
<td>Hall sensor 2</td>
<td>Hall sensor 2 input</td>
</tr>
<tr>
<td>3</td>
<td>Hall sensor 3</td>
<td>Hall sensor 3 input</td>
</tr>
<tr>
<td>4</td>
<td>+5 VDC</td>
<td>Hall sensor supply voltage (+5 VDC: I&lt;=30mA)</td>
</tr>
<tr>
<td>5</td>
<td>GND</td>
<td>Ground</td>
</tr>
</tbody>
</table>
### J4 Encoder

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
<th>Configured Purpose</th>
<th>Encoder Cable (275934)</th>
<th>Head A</th>
<th>Color</th>
<th>Head B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>not connected</td>
<td></td>
<td></td>
<td>1</td>
<td>brown</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>+5 VDC</td>
<td></td>
<td></td>
<td>2</td>
<td>white</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>GND</td>
<td></td>
<td></td>
<td>3</td>
<td>red</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>not connected</td>
<td></td>
<td></td>
<td>4</td>
<td>white</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>Channel A</td>
<td></td>
<td></td>
<td>5</td>
<td>orange</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>Channel A</td>
<td></td>
<td></td>
<td>6</td>
<td>white</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>Channel B</td>
<td></td>
<td></td>
<td>7</td>
<td>yellow</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>Channel B</td>
<td></td>
<td></td>
<td>8</td>
<td>white</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>not connected</td>
<td></td>
<td></td>
<td>9</td>
<td>green</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>not connected</td>
<td></td>
<td></td>
<td>10</td>
<td>white</td>
<td>10</td>
</tr>
</tbody>
</table>

### J5 Digital I/Os

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
<th>Configured Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DigiIN1</td>
<td>PWM - Set Value</td>
</tr>
<tr>
<td>2</td>
<td>DigiIN2</td>
<td>Enable</td>
</tr>
<tr>
<td>3</td>
<td>DigiIN/DigOUT3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>DigiIN/DigOUT4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>6</td>
<td>+5 VDC</td>
<td>Auxiliary output voltage (+5 VDC, &lt;=10 mA)</td>
</tr>
</tbody>
</table>
### J6 Analog I/Os

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
<th>Configured Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AnIN1+</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>AnIN1−</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>AnIN2+</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>AnIN2−</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>AnOUT1</td>
<td>Actual Speed</td>
</tr>
<tr>
<td>6</td>
<td>AnOUT2</td>
<td>Actual Current Averaged</td>
</tr>
<tr>
<td>7</td>
<td>GND</td>
<td>Ground</td>
</tr>
</tbody>
</table>

### Potentiometers P1 / P2

<table>
<thead>
<tr>
<th>Potentiometer</th>
<th>Configured Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td></td>
</tr>
<tr>
<td>P2</td>
<td></td>
</tr>
</tbody>
</table>

### J7 USB

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
<th>Configured Purpose</th>
<th>USB Type A - micro B Cable (403968)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VBUS</td>
<td>USB BUS supply voltage input +5 VDC</td>
<td>Head A</td>
</tr>
<tr>
<td>2</td>
<td>D−</td>
<td>USB Data− (twisted pair with Data+)</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>D+</td>
<td>USB Data+ (twisted pair with Data−)</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>ID</td>
<td>not connected</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>GND</td>
<td>USB ground</td>
<td>5</td>
</tr>
</tbody>
</table>
For Use With Arduino Uno For Testing Straight Motor Commands:

NOTE: An Arduino Uno outputs an 8-bit PWM signal. The motor controller takes a 12-bit PWM signal.

This section of the guide will be utilized to take tests without the use of a MSP430, and instead with an Arduino. This will be without a PID controller, so that data about a system might be collected and a plant model of the system be created. No IMU is attached in this setup.

Example Sketch:
The above sketch shows the communication between the motor controller and the Arduino Uno. This sketch only send a PWM signal to motor and motor controller. This does not include any
form of PID controller or reaction wheel control. This sketch is used for testing the motor to ensure good communication between the motor, motor controller, and Arduino. The settings in the previous section must all be completed first, as they are saved to the motor controller. Set up the motor control software for data collection, as specified above.

- Upload the above sketch onto the Arduino Uno that is being used to control the motor controller.
- Hook up the Arduino, motor controller, motor, and switch as seen above in the pinout diagram.

**Regulation Tuning**

- After setting up the motor and motor controller in the Startup Wizard, Regulation Tuning will automatically pop up.
- Select “Auto-Tuning.”
- Section To Be Completed Soon.
- Click “Start.”
- Test results, similar to the ones seen above, should appear.
- Click “Finish.”
Setting Up the Data Recorder Tool

Upon opening the Data Recording Tool, click “Settings.” The above image will be seen.

The four channels on the left hand side have many options for recording information for the motor:
- Actual Speed
- Actual Speed Averaged
- PWM Input
- Hall Sensor Pattern
- Position Counter Hall Sensor
- Position Counter Encoder
- RC Servo Input
- Analog Input 1
- Analog Input 2
- Potentiometer 1
- Potentiometer 2
- Temperature Power Stage
- External Supply Voltage
- Current Analog Set Value
- Demand Current
For testing with the Arduino Uno, the options selected are:
  ○ Actual Speed
  ○ Actual Current
  ○ PWM Input

The values below these options are to specify how the user wants to plot the data on the Data Recording Tool.

The Data Sampling area of the settings shows how long the user wants to run the test and how many samples will be taken.
  ○ The first two boxes are able to be edited. The last is automatically calculated.
  ○ For this original test, approximately 1 minute of data is desired, so 60,000 ms was inputted into the settings.

Depending upon the set-up of the motor, the section below may be changed.
  ○ If using a switch as outlined above, select “Single Trigger.”
  ○ Select “Digital Signal.”
  ○ Select “Enable” out of the several options.
  ○ For mode, select “Disable → Enable.”
  ○ The time delay may then be decided below. This shows the amount of time it will take until the Data Recording Tool will take record data.

Click “OK.”

The Data Recording Tool is now set up. In order to initiate the data recording, flip the switch.

To export the collected data, right click within the plot and click “Export Recorded Data.”

Example Data:
For Use With Arduino Uno For Testing with IMU (Without PID, to be Added Later)

NOTE: The IMU outputs a 14-bit signal in regards to the gyroscope. The Arduino Uno