

Engduino Protocol



Quick Reference Guide

Protocol.ino is a sketch that allows the Engduino to be programmed or configured with an external program, e.g. Matlab; another hardware device; or a Bluetooth enabled device over the Serial or Serial1 Arduino library interfaces.

For example, you can use Engduino with our Bluetooth module and downloaded with the Protocol.ino sketch to communicate with your Bluetooth enabled mobile phone. You can develop an app that configures your mobile phone to read the temperature sensor on the Engduino board once every second and plot the data on your phone.

Protocol.ino is included in the latest release in the 100.Engduino example folder. You can find the libraries files under the folder [arduinoIDE]\hardare\Engduino\avr\libraries\EngduinoProtocol

Protocol can be used via normal Serial interface (e.g. the Arduino Serial Monitor) or Arduino Serial1 – e.g. hardware serial used by our Bluetooth module.

For quick testing, you can open the Arduino Serial Monitor to test out these commands; or download BlueDuino app on your Android mobile phone to test out the commands.

Engduino Protocol has only tested on Engduino V3. It is not supported in Engduino V1 and V2.

Engduino Protocol Package

The top level Engduino protocol package is surrounded by { } character, with four fields each separated by the ; character. *There is NO space allowed in between the parameters.*

```
{packageType; commandID; parameter 0; parameter 1;.....}
```

- **PackageType**: defines how the package is interpreted by the EngduinoProtocol library codes.
- **CommandID**: defines the commands to be carried out.
- **parameters** (optional): additional parameters passed with the command.

Set Commands

Function	Package type	Command ID	Input Parameters	Input Values	Examples
Set user green led (LED on pin 13)	1	11	<ul style="list-style-type: none"> Status 	1: On; 0:Off	{1;11;1} to turn green LED on {1;11;0} to turn green LED off
Set all LEDs (Colour)	1	10	<ul style="list-style-type: none"> Colour 	0:Red; 1:Green; 2:Blue; 3:Yellow; 4:Magenta; 5:Cyan; 6:White; 7:Off	{1;10;1} to turn all LED Green {1;10;7} to turn all LED off
Set all LEDs (RGB)	1	10	<ul style="list-style-type: none"> R G B 	Value: (0-15) Value: (0-15) Value: (0-15)	{1;10;15;0;15} to turn all LED to magenta
Set individual LED (Colour)	1	10	<ul style="list-style-type: none"> LED num Colour 	Value: (0-15) 0:Red; 1:Green; 2:Blue; 3:Yellow; 4:Magenta; 5:Cyan; 6:White; 7:Off	{1;10;5;2} to turn LED number 5 to colour blue
Set individual LEDs (RGB)	1	10	<ul style="list-style-type: none"> LED num R G B 	Value: (0-15) Value: (0-15) Value: (0-15) Value: (0-15)	{1;10;7;0;15; 15} to turn LED number 7 to colour cyan
Set 16 LEDs different colour	1	10	<ul style="list-style-type: none"> LED0 Color LED1 Color . . . LED15 Color 	0:Red; 1:Green; 2:Blue; 3:Yellow; 4:Magenta; 5:Cyan; 6:White; 7:Off	{1;10;0;0;0;0;0;0;0;0;0;0;0;0;0;0;2} To turn all LED to red except LED 15, which is blue
Set 16 LEDs Colours and Brightness	1	10	<ul style="list-style-type: none"> LED0 colour LED1 colour . . . LED15 colour 	0:Red; 1:Green; 2:Blue; 3:Yellow; 4:Magenta; 5:Cyan; 6:White; 7:Off Brightness: (0-15)	{1;10;3;5;3;5;3;5;3;5;3;5;3;5;3;5;10;10;10;10;10;10;10;10;10;10;10;10;10;10;15;0;} To turn LEDs alternate colour and brightness 10 except LED 14 and 15 (BUG: Not sure why we need to put a ; at the end for it to work.)

Get Commands

Function	Package type	Command ID	Input parameters	Input Values	Input examples	Output example and format
Get temperature data	1	111	<ul style="list-style-type: none"> Sample mode (optional) 	0: Read once (default) -1: Stop continuous sampling if it is active >0: Continuous sampling interval (ms)	{1;111} to take temperature sample once {1;111;1000} To sample temperature sensor at 1 second interval continuously {1;111;-1} to stop sampling	{1;111;22566;1} <ul style="list-style-type: none"> Sample value (in 1/1000th °C) No of averaged samples ¹.
Get accelerometer data	1	112	<ul style="list-style-type: none"> Sample mode (optional) 	0: Read once (default) -1: Stop continuous sampling if it is active >0: Continuous sampling interval (ms)	{1;112} To sample accelerometer sensor once	{1;112;3;27;-988;1} <ul style="list-style-type: none"> X (1/1000th G) Y (1/1000th G) Z (1/1000th G) No of averaged samples
Get magnetometer data	1	113	<ul style="list-style-type: none"> Sample mode (optional) 	0: Read once (default) -1: Stop continuous sampling if it is active >0: Continuous sampling interval (ms)	{1;113} To sample magnetometer sensor once	{1;113;26;469;372;1} <ul style="list-style-type: none"> X (-20000 to +20000) Y (-20000 to +20000) Z (-20000 to +20000) No of averaged samples

¹ If oversampling is set, the value is averaged from the last read/ since oversampling is set with the oversampling rate.

5 Engduino Protocol Quick Reference Guide

Function	Package type	Command ID	Input parameters	Input Values	Input examples	Output example and format
Get light sensor data	1	114	<ul style="list-style-type: none"> Sample mode (optional) 	0: Read once (default) -1: Stop continuous sampling if it is active >0: Continuous sampling interval (ms)	{1;114} To sample light sensor once	{1;114;47;1} <ul style="list-style-type: none"> Light sensor value (0 to 1023) No of averaged samples¹.
Get all sensor data	1	110	<ul style="list-style-type: none"> Sample mode (optional) 	0: Read once (default) -1: Stop continuous sampling if it is active >0: Continuous sampling interval (ms)	{1;110;1000} To sample all sensor values once every second	{1;110;22163;54;7;-1000;-47;324;430;66;1} <ul style="list-style-type: none"> Temperature (in 1/1000th °C) Acc X (1/1000th G) Acc Y (1/1000th G) Acc Z (1/1000th G) Mag X (-20000 to +20000) Mag Y (-20000 to +20000) Mag Z (-20000 to +20000) Light sensor value (0 to 1023) No of averaged samples¹.
Get Button Status	1	120	<ul style="list-style-type: none"> Mode 	0: Read once (default) -1: Disable button interrupt >0: Continuous sampling interval (ms) 1: Package response when button is pressed 2: Package response when button is release	{1;120;2} Button package is sent by Engduino upon button release. <i>Note: You need to use {1;120;-1} to cancel the a previous command</i>	{1;120;2} Package response when button is released

6 Engduino Protocol Quick Reference Guide

Function	Package type	Command ID	Input parameters	Input Values	Input examples	Output example and format
				3: Package response when button is pressed or released	set. Otherwise, it will remains when a new command is issued.	
Get Engduino hardware and protocol version	1	100	none		{1;100}	{1;100;30} First digit: Engduino hardware version 3 Second digit: EngduinoProtocol library version number. This parameter is set in EngduinoProtocol.h
Get Engduino status	1	190	<ul style="list-style-type: none"> Status ID 	0: Number of Oversamples	{1;190;0}	{1;190;0;0} Every sensor reading is the average of 1 sample. (returned number + 1)
Read IR received data	1	160	<ul style="list-style-type: none"> Timeout (us) 		{1;160;1000} to block the code running for 1000us and see if there are data in IR receive buffer	IR received buffer