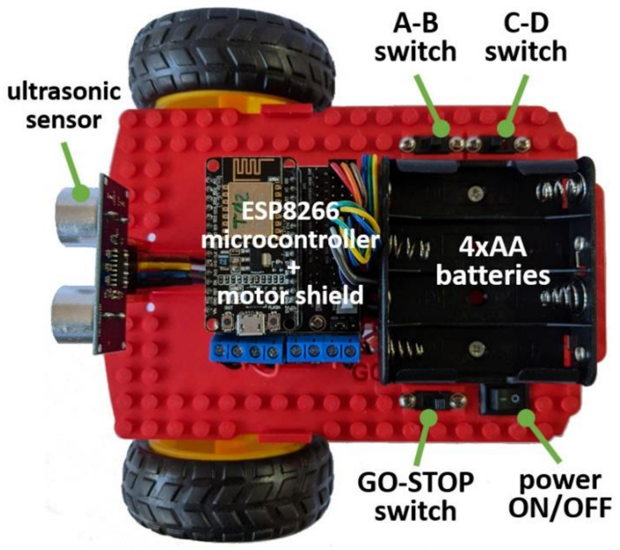
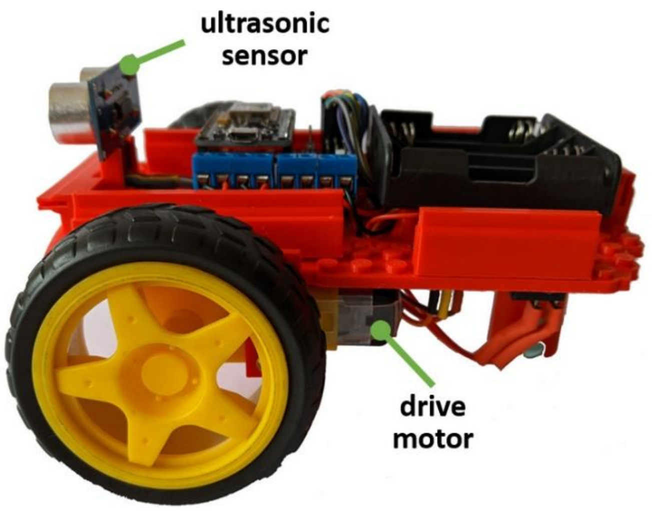
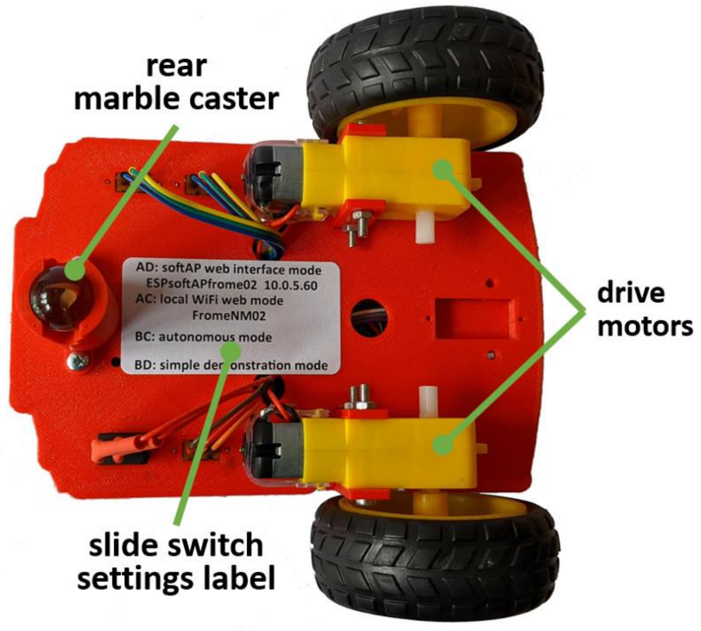
ESP8266 2WD Robot User Guide

Release version 1.1

Construction







AD: softAP web interface mode
ESPsoftAPfrome02 10.0.5.60
AC: local WiFi web mode
FromeNM02
BC: autonomous mode
BD: simple demonstration mode

The *ESP8266 2WD Robot* uses two geared drive motors 'underslung' on a 3D printed chassis plate that has Lego-compatible studs on the spare space on the top surface. The drive wheels are at the front of the robot with a 'marble castor' assembly at the rear and an ultrasonic sensor fitted on the top/front of the robot.

Control for the Robot is provided by an ESP8266 microcontroller (packaged as a NodeMCU v1.0 module) connected to a L293D ESP-12E motor drive module.

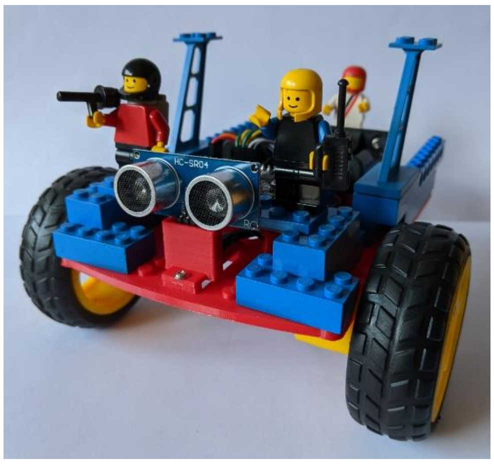
Both the microcontroller and the drive motors are powered by 4xAA (high power, rechargeable) batteries where their combined power supply is controlled by an ON/OFF rocker switch.

Logical operation is controlled by three slide switches: A-D, C-D, and GO-STOP, which give four possible mode settings (AD, AC, BC, and BD) that are set to operate with the GO-STOP switch.

Customisation

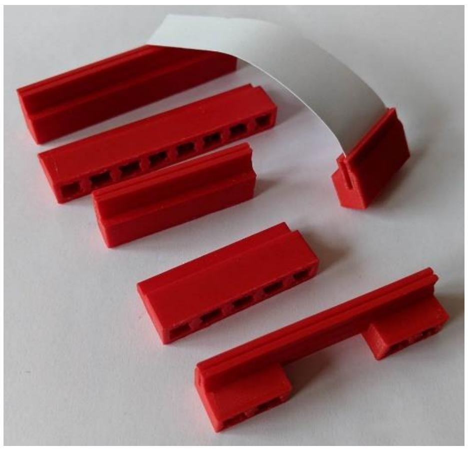
The *ESP8266 2WD Robot* has been designed specifically to allow it to be customised in a number of different ways as described below.

Lego builds



The Lego-compatible studs on the top surface of the 3D printed chassis plate allow the robot to be customised in any way imagined with standard Lego bricks and other Lego components as illustrated in the example build above.

Card add-ons



An alternative customisation method is to use some special 3D-printed, Lego-compatible 'card connectors' as shown in the image on the right.

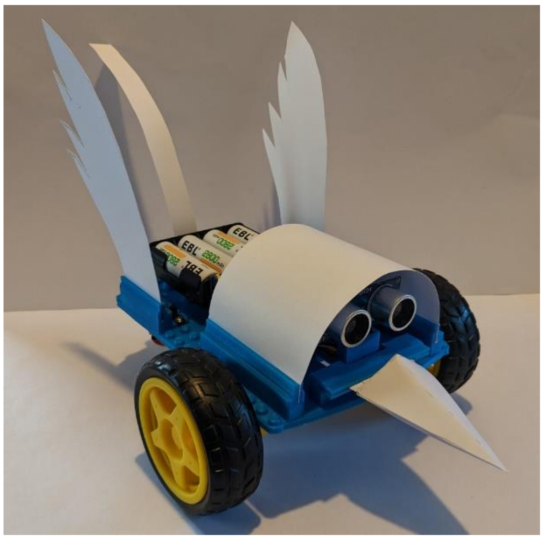
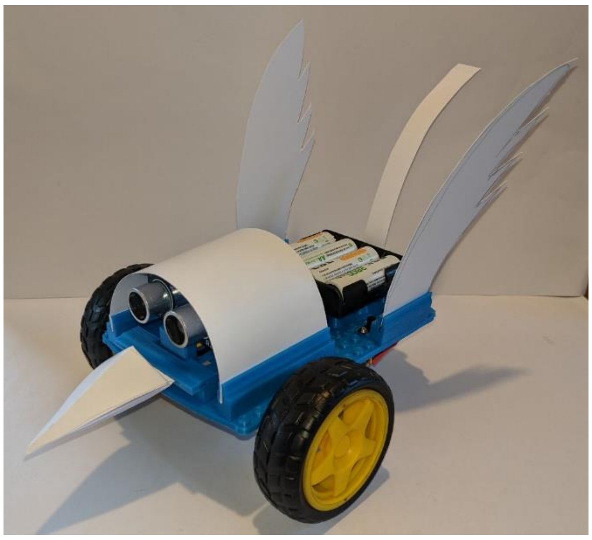
These connectors can be attached to the top surface of the chassis plate, just like Lego bricks, but can then have sections of card 'slotted' into them.

A lightweight card (200 gsm) works best, which provides sufficient stiffness to create interesting build effects, whilst being held securely in the Lego connector slot.

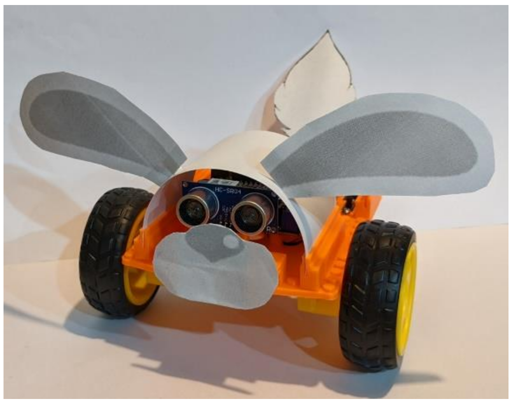
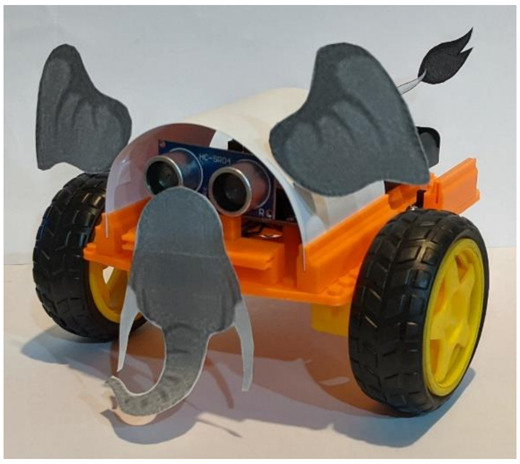
When first used the connector slot may be a little tight and might need to be gently opened with a small screwdriver.

Use of card cut-outs of different shapes allows a very wide variety of builds to be created that are only limited by your imagination!

The images below show an example build where wings, a tail, and a beak have been added along with a shrouded front section that makes a 'body' with the ultrasonic sensor appearing like a pair of eyes.

The card elements used in the build example above have been cut-out on a free-hand basis, but a number of plain templated cut-outs that can be coloured in, along with more detailed templates for animal ears, tails, noses/faces, etc., are also being developed as illustrated below.

Operation

Before inserting the AA batteries into its holder make sure the power ON/OFF rocker switch is in the OFF (O) position and the GO-STOP slide switch is in the STOP position.

Start-up will take a couple of minutes, but (at the moment!) there is not any indication that the system has fully started, so leave it at least a couple of minutes before using the A-B and C-D slide switches to select one of four possible operational modes. Once a mode is selected it will not start operating until the GO-STOP slide switch is moved to the GO position.

The four possible operational modes are as follows:

A plus D positions: starts what is called the 'softAP' web interface mode which is useful for using the more extensive operational web interface when either there is not any locally available WiFi, or it is out of range, or its access details are not known.

- the 'softAP' web interface mode makes the robot into its own custom WiFi Access Point where the usual WiFi broadcast name (SSID) format is: ESPsoftAPxxxxxx where xxxxxx can be hard coded to an individual value for each robot so that several of them can operate in the same area; the specific xxxxxx value should be shown on the label that can be fixed to the underside of the robot;
- for all robots the WiFi password (WEP key) is the same and is: *pswd12345*
- this mode also automatically starts a web server which can be accessed at the 10.0.5.NN IP address i.e., use **http://10.0.5.NN** as the URL, where NN is an individual hard coded number for each robot, again so that several of them can operate in the same area, where this number should also be shown on the label fixed to the underside of the robot;
- to use this mode the access device (tablet, 'phone, etc.) should be connected to the WiFi broadcast name, i.e., ESPsoftAPxxxxxx and then the IP address accessed.

A plus C positions: starts what is called the 'local' WiFi web mode, where it is assumed that the local WiFi access credentials have been input as one of the five optional WiFi connections stored in the system (details described later) so that use in a number of different locations can be preconfigured.

- the system cycles through the five optional WiFi connections stored in the system and attempts to connect to the first option that it 'discovers' is being broadcast locally;
- a web server is automatically started that assumes it can be assigned a dynamic IP address from the local WiFi's router (DHCP mode) but where a unique *hostname* is also set so that several of them can operate in the same area, where this *hostname* is also shown on the label fixed to the underside of the robot;
- to use this mode the access device (tablet, 'phone, etc.) should connect to the local WiFi and then the **http://*hostname*** address accessed.

B plus C positions: starts the 'autonomous' mode, which will start the robot going in a forward direction, at the currently set default speed (see later), once the GO-STOP switch is set to GO. The ultrasonic sensor operates in a continuous mode and when it detects an obstacle within the currently set value for the "*take avoidance action distance (cm)*" it triggers a "*stop and go in a different direction*" action.

B plus D positions: starts the 'demonstration' mode, which runs through a defined series of up to 20 demonstration actions which consist of *command, duration* pairs.

The command options are the following:

FWD	STOP	SPINR	TURNR
BACK	DELAY	SPINL	TURNL

.. and the duration is a time in milliseconds.

For all four modes once a mode has been selected it is then made to operate by sliding the GO-STOP switch across to the GO position.

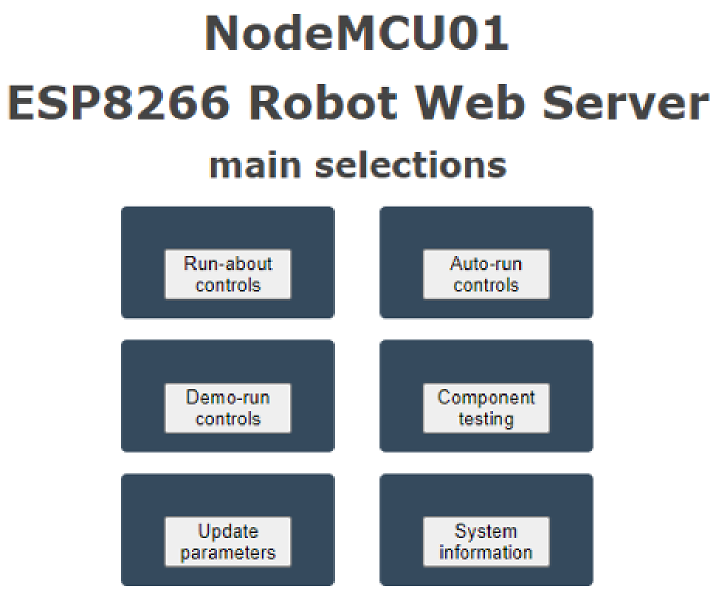
Moving the slide switch back to the STOP position at any time will stop the current operation mode (although it may have to finish a specific action first) and then any alternative mode option can be selected using the A-B and C-D slide switches.

Web interface

When either the 'softAP' web (A-D) or the 'local' WiFi web (A-C) modes are used, an extensive web interface is available if a 'phone, tablet, etc., also connects to the appropriate WiFi signal, i.e. either the 10.0.5.NN IP address for the 'softAP' option, or your local WiFi which must also have been set up in the WiFi options (discussed later).

The web interface that is displayed provides the following options:

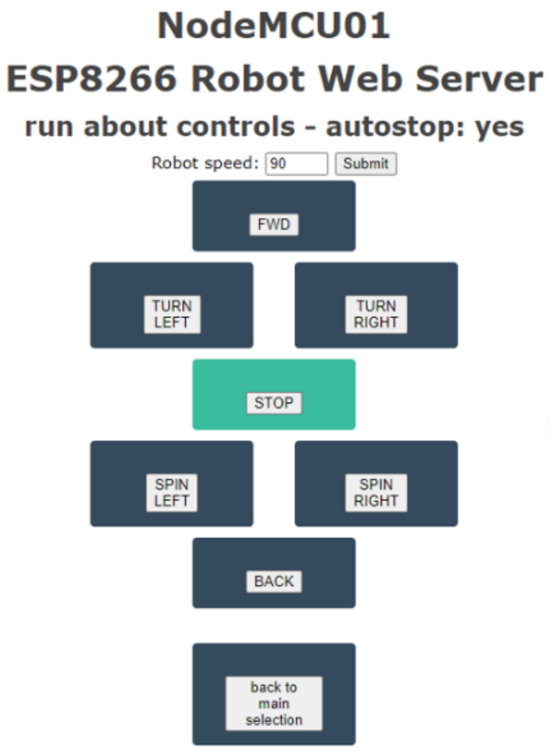
- **main selections**: shows the default top-level display that allows all the various 'main' options to be selected, as shown in the screen shot below, by clicking/tapping an individual button:


NodeMCU01

ESP8266 Robot Web Server

main selections

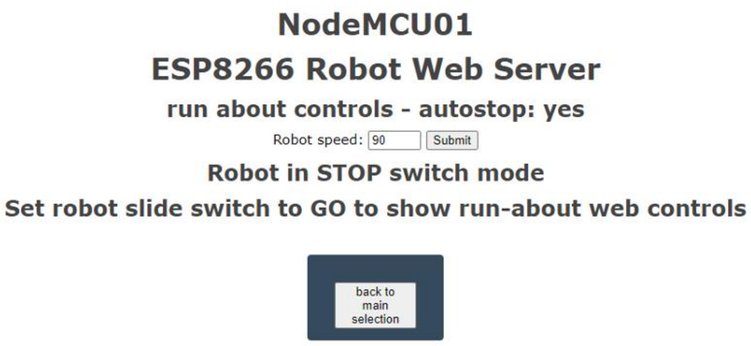
- **Run-about controls**: if the position of the GO-STOP switch is GO a 'control' set of buttons is displayed, as shown below left, that when clicked/tapped will signal the robot to go forward, back, turn, spin, or stop. Alternatively, if the GO-STOP switch is on STOP the display is as shown below right. It should be noted however that as the robot 'control' signals are sent via WiFi, they may experience some time lag, so a robot stop control signal may not be received quickly enough to avoid a crash. The system therefore has an 'autostop' function that uses the ultrasonic sensor to detect an obstacle when going forwards and automatically stops the robot if it is within a default distance of an obstacle, irrespective of the current robot 'control' signal.


NodeMCU01

ESP8266 Robot Web Server

run about controls - autostop: yes

Robot speed: 90


NodeMCU01

ESP8266 Robot Web Server

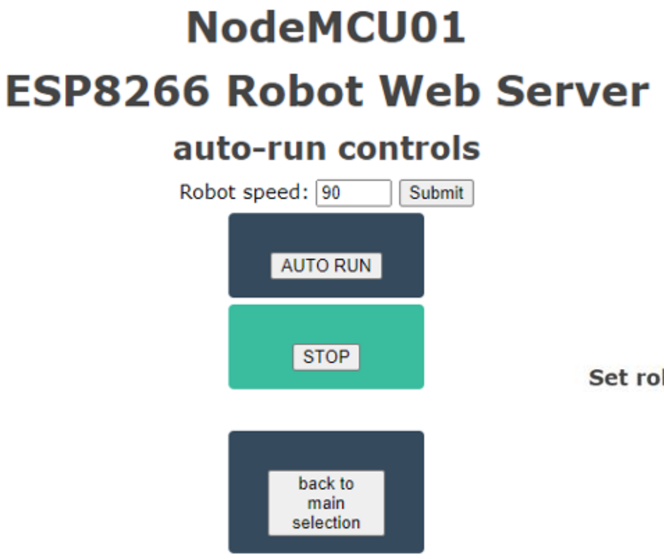
run about controls - autostop: yes

Robot speed: 90

Robot in STOP switch mode

Set robot slide switch to GO to show run-about web controls

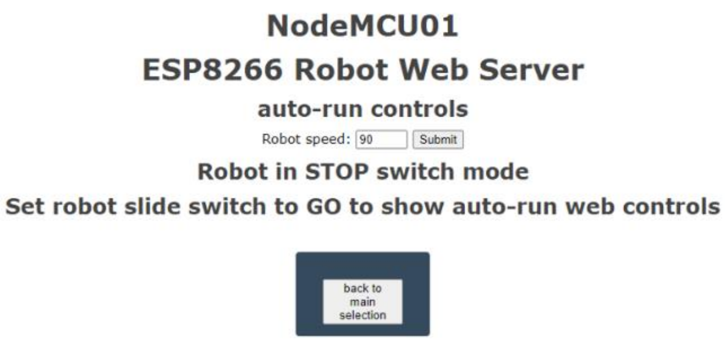
- **Auto-run controls**: a simple AUTO RUN/STOP interface that puts the robot into a similar 'autonomous' mode of operation as is provided by the 'B plus C' slide switch setting, with the alternative screens for the GO-STOP switch settings as shown below:


NodeMCU01

ESP8266 Robot Web Server

auto-run controls

Robot speed: 90


NodeMCU01

ESP8266 Robot Web Server

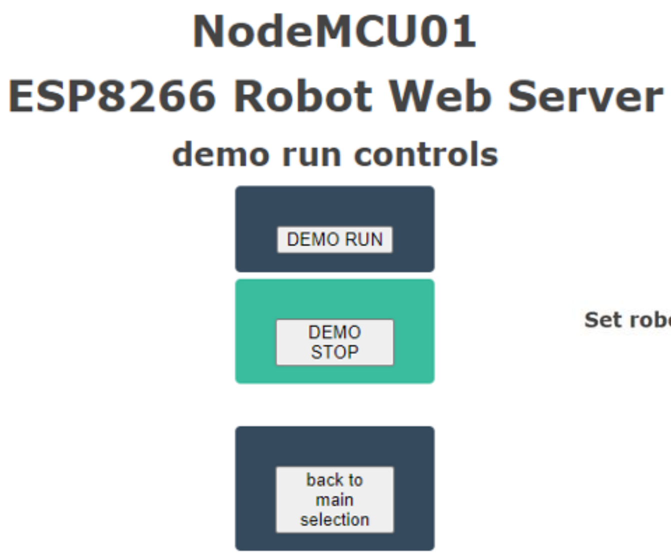
auto-run controls

Robot speed: 90

Robot in STOP switch mode

Set robot slide switch to GO to show auto-run web controls

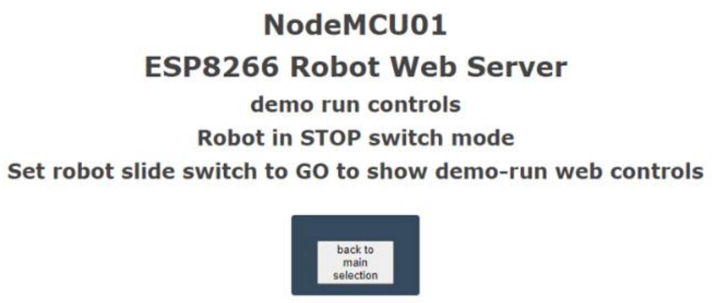
➢ **Demo-run controls**: a simple DEMO RUN/DEMO STOP interface that puts the robot into a similar demonstration mode of operation as is provided by the 'B plus D' slide switch setting, with the alternative screens for the GO-STOP switch settings as shown below:



NodeMCU01

ESP8266 Robot Web Server

demo run controls



NodeMCU01

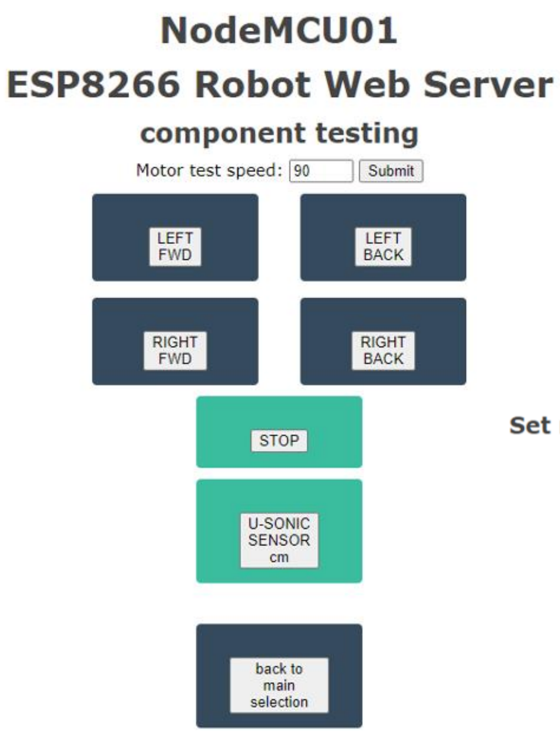
ESP8266 Robot Web Server

demo run controls

Robot in STOP switch mode

Set robot slide switch to GO to show demo-run web controls

➢ **Component testing**: allows individual 'components of the robot to be separately tested which is useful if something does not seem to be working correctly, with the alternative screens for the GO-STOP switch settings as shown below:

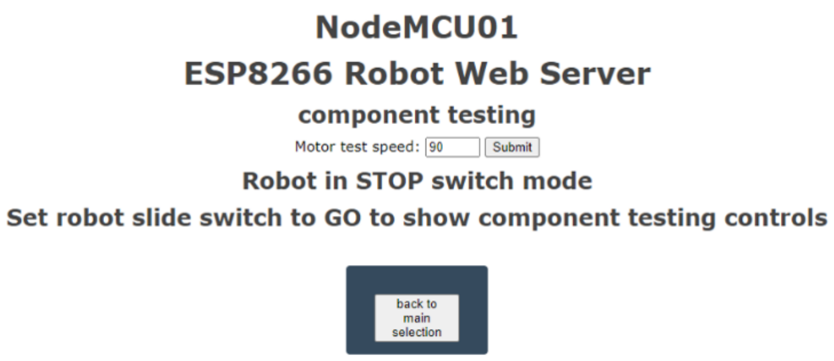


NodeMCU01

ESP8266 Robot Web Server

component testing

Motor test speed: 90



NodeMCU01

ESP8266 Robot Web Server

component testing

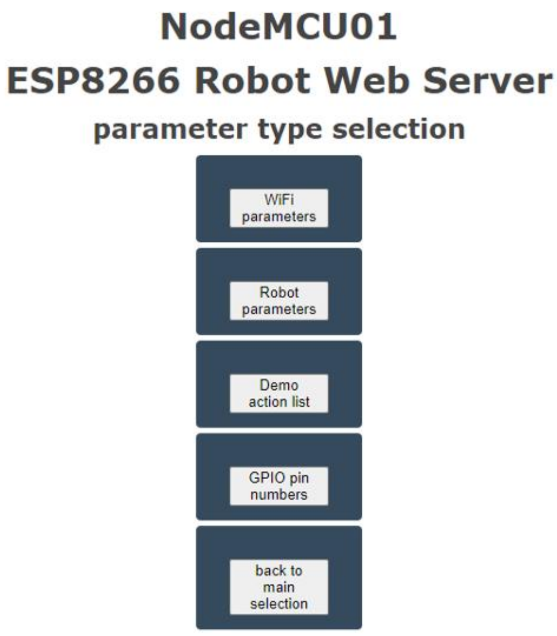
Motor test speed: 90

Robot in STOP switch mode

Set robot slide switch to GO to show component testing controls

- **Update parameters**: allows sets of internally used parameters to be updated that are then stored for future use.

  The initial screen, as shown below, allows you to select which set of parameters are to be updated.



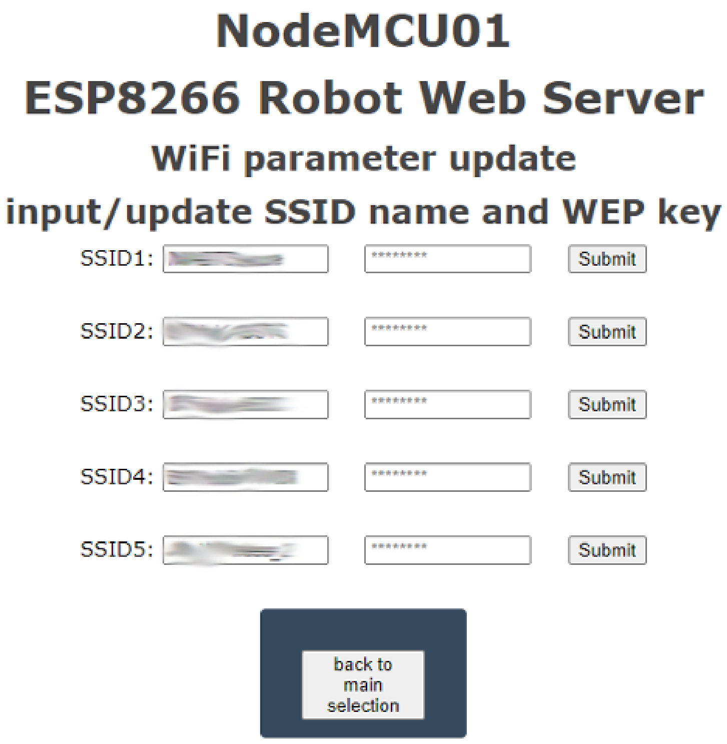
NodeMCU01

ESP8266 Robot Web Server

parameter type selection

  Each subsequent option is described next.

  - ***WiFi parameters***: allows the access credentials for up to five optional local WiFi details to be input or updated, as shown in the screen shot below, which as you see will never show the passwords, so take great care when entering these.



NodeMCU01

ESP8266 Robot Web Server

WiFi parameter update

input/update SSID name and WEP key

SSID1: ******** Submit

SSID2: ******** Submit

SSID3: ******** Submit

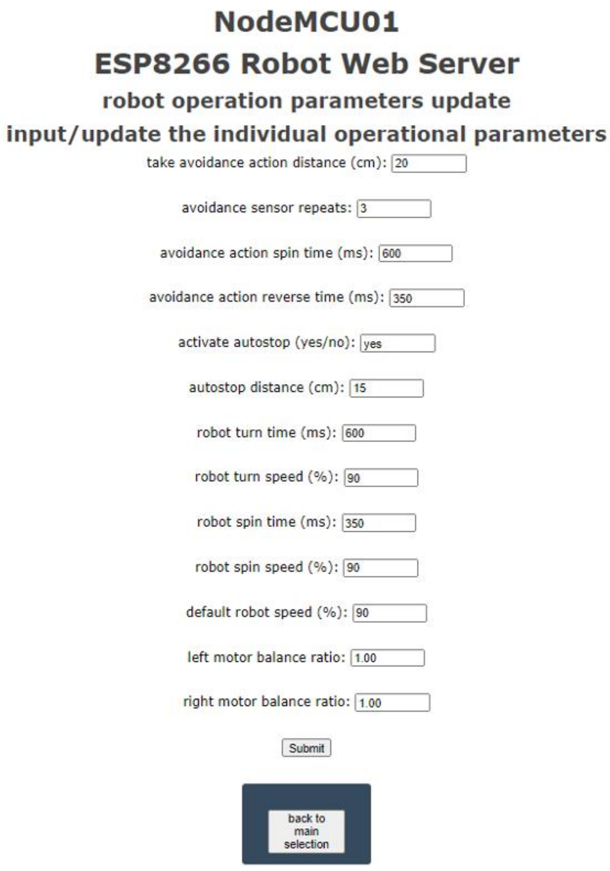
SSID4: ******** Submit

SSID5: ******** Submit

- ***Robot parameters***: allows various robot control parameters, such as the *"take avoidance action distance (cm)"* parameter, to be updated, as shown in the screen shot on the right.

  The main parameters that you may need to adjust are:

  - the left or right 'motor balance' ratios which allow you to compensate for the robot drifting to either the right or left when it is supposed to be going in a straight line,
  - the 'take avoidance action distance' which adjusts the ultrasonic sensed distance that is used to stop the robot and to turn in a different direction, and
  - the avoidance action spin time' which is the time in milliseconds that the robot will take when spinning to either the left or right when executing an avoidance action.


NodeMCU01
ESP8266 Robot Web Server
robot operation parameters update
input/update the individual operational parameters

take avoidance action distance (cm):	20
avoidance sensor repeats:	3
avoidance action spin time (ms):	600
avoidance action reverse time (ms):	350
activate autostop (yes/no):	yes
autostop distance (cm):	15
robot turn time (ms):	600
robot turn speed (%):	90
robot spin time (ms):	350
robot spin speed (%):	90
default robot speed (%):	90
left motor balance ratio:	1.00
right motor balance ratio:	1.00

- **Demo action list**: allows up to 20 demonstration *command, duration* pairs to be input or updated, as shown on the right.

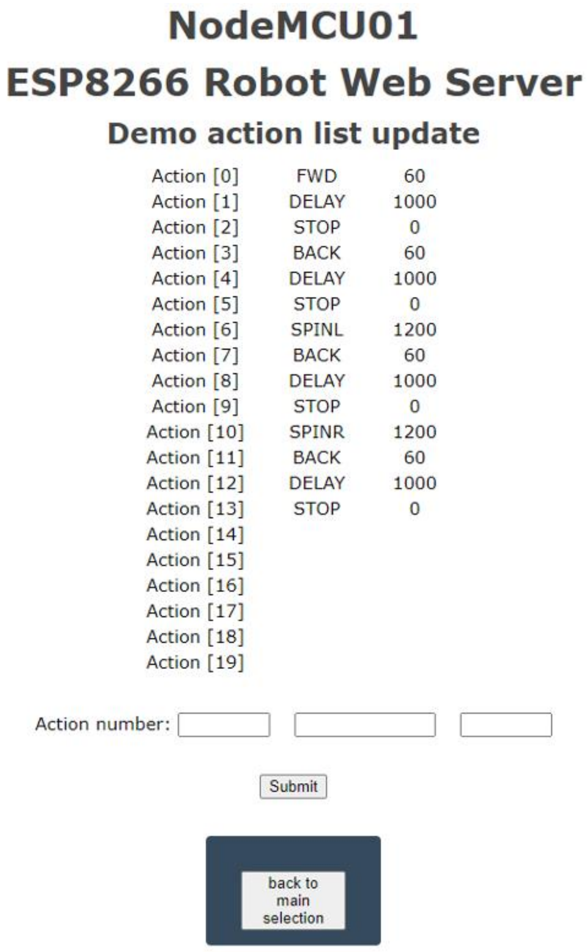
  Use the form part of the display to input an 'Action number' with its associated command and duration and click the submit button.

  The command options are the following:

FWD	STOP	SPINR	TURNR
BACK	DELAY	SPINL	TURNL

.. and the duration is a time in milliseconds.

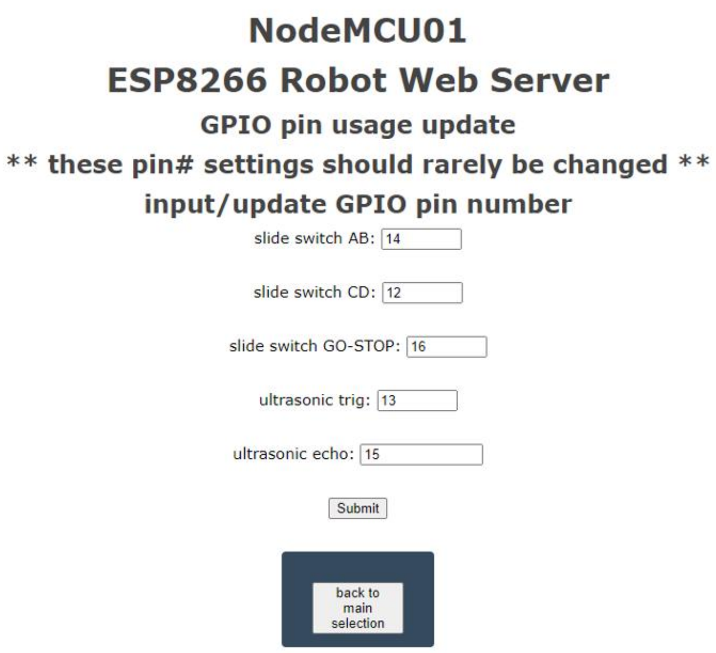
It should be noted that it is not necessary to 'use up' one of the action steps to make a STOP unless you want the robot to stop for a period of time, i.e., you can go straight for a one movement command to another without a STOP in between.


NodeMCU01
ESP8266 Robot Web Server
Demo action list update

Action [0]	FWD	60
Action [1]	DELAY	1000
Action [2]	STOP	0
Action [3]	BACK	60
Action [4]	DELAY	1000
Action [5]	STOP	0
Action [6]	SPINL	1200
Action [7]	BACK	60
Action [8]	DELAY	1000
Action [9]	STOP	0
Action [10]	SPINR	1200
Action [11]	BACK	60
Action [12]	DELAY	1000
Action [13]	STOP	0
Action [14]		
Action [15]		
Action [16]		
Action [17]		
Action [18]		
Action [19]		

Action number:

- ***GPIO pin numbers***: not normally used but allows some internal system references to be changed if a substantially different physical build is subsequently used, as shown on the right.



NodeMCU01

ESP8266 Robot Web Server

GPIO pin usage update

** these pin# settings should rarely be changed **

input/update GPIO pin number

slide switch AB: 14

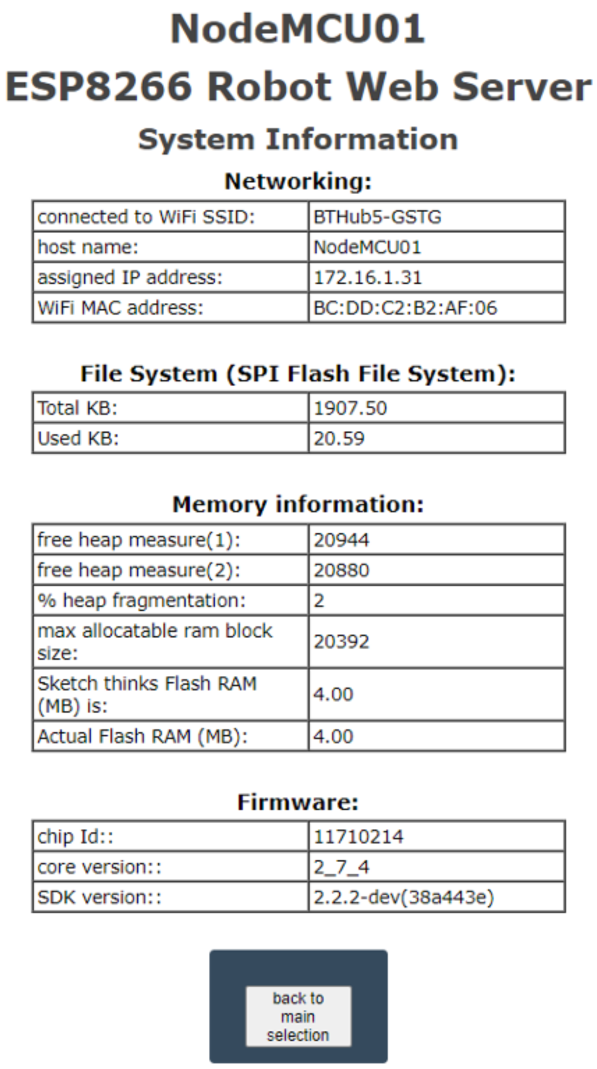
slide switch CD: 12

slide switch GO-STOP: 16

ultrasonic trig: 13

ultrasonic echo: 15

- **System information**: displays a set of technical information for the current '*state'* of the robot, as shown below:



NodeMCU01

ESP8266 Robot Web Server

System Information

Networking:

connected to WiFi SSID:	BTHub5-GSTG
host name:	NodeMCU01
assigned IP address:	172.16.1.31
WiFi MAC address:	BC:DD:C2:B2:AF:06

File System (SPI Flash File System):

Total KB:	1907.50
Used KB:	20.59

Memory information:

free heap measure(1):	20944
free heap measure(2):	20880
% heap fragmentation:	2
max allocatable ram block size:	20392
Sketch thinks Flash RAM (MB) is:	4.00
Actual Flash RAM (MB):	4.00

Firmware:

chip Id::	11710214
core version::	2_7_4
SDK version::	2.2.2-dev(38a443e)

ESP8266 software

The ESP8266, packaged as the NodeMCU v1, provides a microcontroller with a reasonably large amount of non-volatile memory. This means that, like most microcontrollers, whilst you can only 'load it' with the one set of code, that code is always available, and you just need to power the module for the code to start running.

The code for the *2WD ESP8266 Robot* has been developed using the Arduino IDE and because the NodeMCU can also use Flash memory as if it were a file system, all the various parameters managed through the web 'Update parameters' code are stored in this Flash memory and can therefore be updated and re-read whenever the code is started (which is one of the reasons it does take a little while to start up!).

The current code has undergone several cycles of revision, and further updates are possible since all the available memory is still not being used, so there is more 'head room' for more functionality.

Feedback of any kind is therefore welcome in order to identify ways of further improving the operational 'robustness' of the robot and to provide more options and features.