

## jmMotor GUI Module

This is a graphical user interface for jmCLIG firmware module jmMotor.c

The firmware module must be compiled and loaded into the chip before using the graphical user interface.

The firmware can be used with a standard communication application, like HyperTerminal, but jmMotor is more user friendly and limit access only to pins on MBED prototyping board (DIP5 to DIP30 and LED1 to LED4).

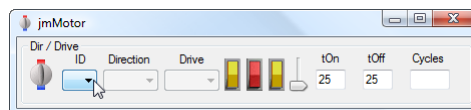
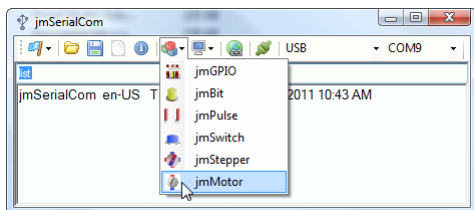
This module can start up to 4 different processes for controlling direction and rotation speed of DC motors by software PWM on MBED DIP pins and LED1 to LED4.

This module automatically sets the selected pins as an output pins and writes continually to the drive output pin, when the motor does not run at 100% of its maximum speed. There is no need to configure output pin direction before using it.









Care should be taken when you turn a microcontroller pin into an output pin. You should make sure that nothing is wired to that pin that can make a short circuit to ground or a short circuit to the supply. Beware of pins previously used as input, because these are potential short circuit hazards if external hardware is still connected to those pins.

Start jmMotor GUI Module from jmSerialCom



### How to

-  Select ID
-   Select Direction and Drive pins that will be wired to the hardware
-  Hit a yellow switch to start the motor in a direction
-  Hit the red switch to stop the motor
-  Change slider position, wheel mouse, arrow keys or Edit Delay value to change speed.
- Change Cycles value to pulse the motor a fixed amount of pulses.

You first select a process ID, the control will send a message asking a report for that process ID from the chip. If this process is initialized, the control will be updated with the pins wired to that process and animation will be updated according to the running process.

If the process is not initialized, no pin will be selected.

You initialize or change the motor parameters by selecting where the hardware controller is wired to the chip by selecting controller direction and drive pins.

You hit a yellow switch to start the motor in one direction.

You hit the red switch to stop the motor.

When the motor is running, you can change the speed by using the slider, wheel mouse, arrow keys or by editing tOn, tOff or both values.

### tOn, tOff and Cycles

tOn, tOff editing boxes can modify the speed.

An empty Cycles editing box value or a value of 65535 will run the motor continuously. Other values of Cycles will run the motor for that duration of cycles.

**After editing tOn, tOff or Cycles values, you must hit a yellow switch to send the command.**

A tOn value of 0 will stop the motor.

A tOff value of 0 will run the motor at 100% PWM if tOn value is not 0.

### Slider



The slider changes tOn et tOff values more rapidly than edit boxes. On and OFF values for the slider are limited to (0..20). Edit Boxes can vary values from 1 to 255.

You can use up and down arrow keys to move the slider more precisely.

You can use wheel mouse as well to vary the slider.

**Slider act as a speed controller only when after the motor is started.**

Each slider action sends a message to the chip, the chip will also return a message for each change. You will see these messages in jmSerialCom text zone.

### Processes

Up to 4 different motors can be controlled at the same time.

### Killing a Process

When you start a continuous jmMotor process in the chip, if you close the GUI control, the process continue to run in the chip.



You kill a process by hitting the Off button for that process.

## Running Process ?

Processes can be running on your chip when you start a jmMotor GUI.

When you select an ID for that control, the control query the chip for that process ID and the chip will return the status of that process.

- If the process is not initialized, no pin will be selected.
- if a process is assigned to that ID, the animation will show the status the process is in and the output pins connected to that process. Cycles values are not reported back.

## Changing Process

You can always overwrite a process by giving it new values and hitting a yellow switch.

## Animation

This GUI control simulates a motor. This animation is controlled by chip reports and if you disconnect the chip while the GUI control is still running it won't be updated anymore. The animation provides direction of rotation when the motor runs.

## Starting Sequence

Always connect your chip first, then start jmSerialCom. This way auto connect feature of jmSerialCom will automatically connect to the chip if you use the same USB slot. Changing USB slot changes Virtual Com Port Number (Windows way of doing).

## Stopping Sequence

You can chose to kill or not a process before closing your GUI controls. Closing jmSerialCom closes all GUI controls. If the chip is still powered, all started non killed processes will continue to live in the chip.

jmMotor GUI communicates with on chip state machine motor controllers.



## Motor Animation

Animation is provided to show that the motor is running. Direction is simulated but not speed.

It is up to you to wire your motor in a way that the animation and your motor turn in the same direction.