The critterOS Panels Explained

To be able to offer a dynamic experience where different sensors and actuators can be attached to the Firefly board, a panels system with an interpreter was devised to deal with the complexity of the many use cases. Here follows an explanation of the different panels that can be created on the critterOS web interface.

To create a new panel use the **New Panel** button in the System Panels section. The New Panel button brings up the New Panel dialog where you can select the name, type, driver and pins.

[SCREENSHOT: New Panel dialog showing type selector, hardware dropdown, and pin fields]

List of User Panels

Input Panel (panel_in)

This is a general container that can accept many inputs and provides a calibration interface. This panel needs to specify the driver to use with it depending on the input hardware used.

Input Interfaces:

- **Analog In**: Analog input (0-3.3V)
- **Digital In**: Digital input handling
- **DS18B20**: Temperature sensor (OneWire)
- **BMP180**: Atmospheric pressure + temperature (I2C)
- **BME280**: Pressure + temperature + humidity (I2C)
- **BME680**: Air quality - pressure + temp + humidity + gas (I2C)
- **MPU6050**: Accelerometer (I2C)
- **IR Remote**: Infra Red receiver (NEC protocol)
- **US Sensor**: Ultrasonic distance (HC-SR04)
- **Flow / RPM**: Pulse counter for flow sensors, RPM
- **HX711**: Load cell (weight sensor)
- **DHT22**: Temperature + humidity (digital)
- **DHT11**: Temperature + humidity (budget digital)
- **VL53L0x**: Time-of-flight distance (I2C)

- **SHT31**: Temperature + humidity (I2C, high accuracy)
- **BH1750**: Light sensor (I2C, lux)
- **UDP Remote**: Subscribe to a panel on another board

Once a Driver is selected it is necessary to specify one or two pins depending on the kind of input. I2C sensors use SDA (D1) and SCL (D2).

[SCREENSHOT: Input panel card showing values, calibration buttons, and driver debug info]

Output Panel (panel_out)

Similar to panel_in but for outputs. You can input the desired output or use the panel title as a variable in programs.

Output Interfaces:

- **Digital Output**: For controlling relays, LEDs, actuators, etc.
- **Servo**: For controlling servo motors (0-180 deg)
- **PWM**: 0-100% duty for pumps, valves, dimmers
- **Stepper**: STEP + DIR driver board protocol
- **WS2812**: NeoPixel / addressable LED

[SCREENSHOT: Output panel card showing toggle and value]

Set Panel

Its use is to attach a Digital Output to this panel. It can be controlled by the web interface and can also be switched on and off by other panels or rules.

Timer Panel

This panel can be attached to any digital output that needs timing functionality.

Timer mode (schedule)

In this mode the numbers entered indicate the period in seconds that the output is off and on respectively. For example, `5-5` means 5 seconds off, 5 seconds on, repeating. If repeat is left blank or 0 means forever, otherwise you can set how many repetitions before it stops. You can enter more than one pair of numbers for periods of off and on times, for example `3600-60-30-120` every hour will turn on for 60 seconds, off for

30, and on again for 120 seconds.

The Output selector can also be used to manually switch the output on or off.

[SCREENSHOT: Timer panel in schedule mode showing Off-On field with "5-5" and Count limit]

Hours mode (time of day)

Switching the Timer/Hours selector to the Hours position puts the timer in Time of the Day mode. In this case you input the On and Off absolute times for the output (e.g., 08:00 / 20:00). The output follows the on/off time window.

[SCREENSHOT: Timer panel in hours mode showing On time and Off time fields]

Push Button Panel

This panel manages physical push buttons and allows binding 3 different triggers (click, double-click, long press) with any virtual button from another panel to trigger that action.

The released state selector allows accounting for different configurations for push buttons tied to +Vcc or GND. Select the appropriate released state for the push button to function properly.

Once you click the select button corresponding with the action you want to set, the cursor becomes a crosshair and the next button or panel you click on is assigned to that particular push button action.

[SCREENSHOT: Push button panel showing released state selector and 3 binding slots]

Program Panel (Automation)

This panel has a text box with a simple interpreter that can process a very basic set of instructions. Keywords that can be used:

- **Numbers**, **Panel names**, **User Variables**
- **Arithmetic:** + - * / ( ) =
- **Binary:** & |
- **Comparison:** > < is
- **Conditional:** if else

When the run selector is ON, the program executes every 50ms. To make a program run just once, set the last line to the program name = 0 (e.g., `task=0;`).

Program panels can also start other program panels by setting their value to 1. This allows for complex logic beyond what a single program can achieve.

[SCREENSHOT: Program panel showing code editor with IF/THEN rules and program selector]

Flow Control Panel

This panel can manage any kind of sensor that relates electrical pulses with a physical quantity, for example odometers, flow sensors, etc.

It's a Panel_in with a flow Driver. For the calibration process: Reset the counter (clicking on the R button), then bring the counter up to a known quantity (e.g., fill a container with a known capacity), press the calibration High button and input that capacity.

The second value is the live rate in quantity/min (e.g., L/min), scaled by the same calibration slope.

[SCREENSHOT: Flow control panel showing total, rate, status, limit, and calibration buttons]

Dispenser Panel

This is a special panel that acts as an automatic dispenser. On creation, this panel takes an input (weight/level sensor) and an output (PWM pump/valve). The PWM output goes to 100% on Start. Upon reaching 90% of the Limit, it eases to a PWM setting between 100% and the min set level. Once it reaches 95%, it goes to the min set level. This avoids overshooting.

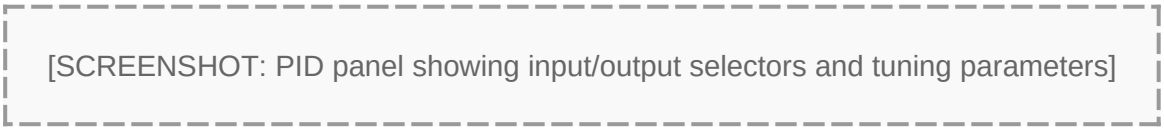
To set the min speed, type the speed in the Output edit box and press Set Min.

Parameters: Limit (target quantity), Min % (minimum PWM), Max % (maximum PWM), Slowdown (seconds for gentle-fill ramp).

[SCREENSHOT: Dispenser panel showing progress, duty, and parameter inputs]

PID Controller Panel

PID controller reads an input panel and drives an output panel. Anti-windup, derivative on measurement. Parameters: setpoint, Kp, Ki, Kd, output min/max.

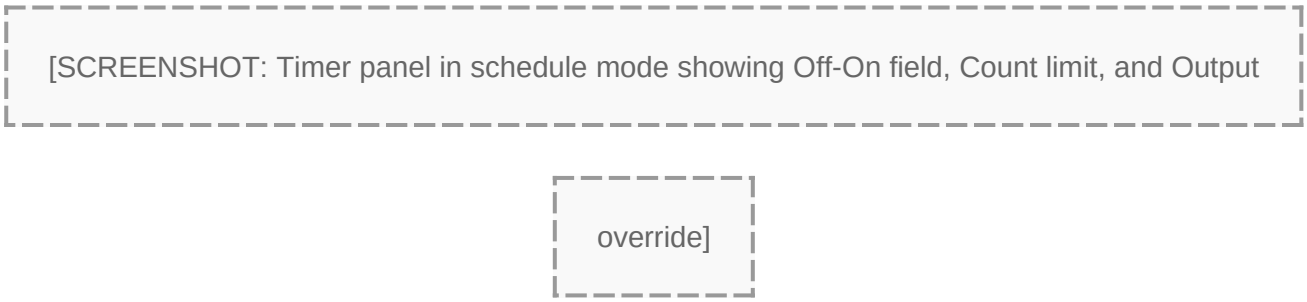


Timer Panel

This panel can be attached to any digital output that needs timing functionality.

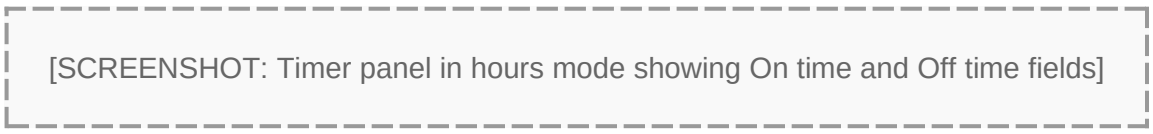
Schedule mode

In this mode the numbers entered indicate the period in seconds that the output is off and on respectively. For example, `5-5` means 5 seconds off, 5 seconds on, repeating. If count limit is left blank or 0 means forever, otherwise you can set how many repetitions before it stops. You can enter more than one pair of numbers for periods of off and on times, for example `3600-60-30-120` every hour will turn on for 60 seconds, off for 30, and on again for 120 seconds.



Hours mode

Switching the Timer/Hours selector to the Hours position puts the timer in Time of the Day mode. In this case you input the On and Off absolute times for the output (e.g., 08:00 / 20:00). The output follows the on/off time window.



Telegram Panel

Sends messages to a Telegram chat from rules or panel events. Configure a message template (supports `{device}` and `{value}` placeholders) and an optional source panel.

Number Panel (Setpoint)

Editable setpoint value, usable in rules and PID controllers. The value can be changed from the dashboard or via MQTT.

UDP Remote Panel

A panel can subscribe to a panel on another board over WiFi. Create an input panel with the `udp_remote` driver, enter the remote board MAC address and the remote panel name. Multi-value panels support sub-values: use `MAC/panel.value` notation (e.g., `C8:C9:A3:64:80:C8/press.temperature`) to follow a specific reading. The remote value updates automatically via UDP broadcasts.

[SCREENSHOT: New Panel dialog showing UDP remote configuration with MAC and remote panel dropdown]

Multi-value panels

Many sensors publish more than one reading. Instead of creating a separate panel for each value, critterOS uses a **values map** with named keys:

- `""` (empty string) = the primary/default value
- `"temperature"`, `"pressure"`, `"humidity"`, `"gas"`, `"lux"`, etc. = named sub-values

For example, a BMP180 panel publishes `{"": 1013.25, "pressure": 1013.25, "temperature": 19.5}`. UDP remote panels can subscribe to a specific sub-value using `MAC/panel.value` notation.

I2C scanner

Available at `/i2cscanner.html`, linked from the dashboard. Scans the I2C bus for connected devices and reports their addresses.

Every input panel includes

- **Calibration**: offset/factor or two-point (Low/High) calibration
- **Averaging**: optional 16-sample moving average
- **Logging**: record values to a CSV file on the board
- **Display**: show on the OLED (per-panel toggle)
- **MQTT**: publish state to a broker or Home Assistant
- **UDP**: publish to other boards on the LAN

All of this runs on a $4 Wemos D1 Mini (ESP8266) or ESP32 board, with no server, no cloud and no subscription.