

# Overview

## Purpose

- To explore using the Tinkerforge Microcontroller C++ Bindings with setup
  - Arduino & IO4 Bricklet
  - ESP32 & IO4 Bricklet
- To blink an LED (IO4 Channel 0, Output), 10 times [*io4led*]
- To switch an LED (IO4 Channel 0, Output) via push-button (IO4 Channel 1, Input) [*io4ledbtn*]
- To switch an LED (IO4 Channel 0, Output) via Serial Line [*io4edserial*]

## Parts List

- Arduino UNO & MEGA, ESP-WROOM-32
- Tinkerforge IO4 Bricklet V2.0, Breakout Bricklet V1.0, 7p-10p cable
- DFRobot Digital Piranha Blue LED Module V2
- DFRobot Digital Push Button Red V2
- Various wires

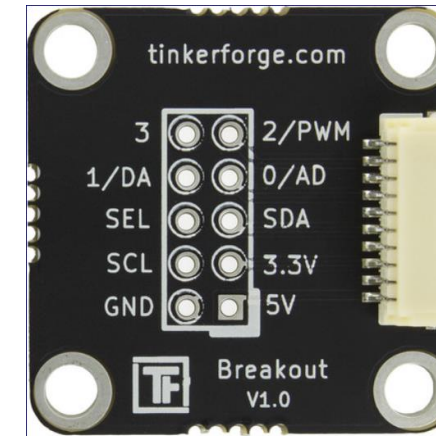
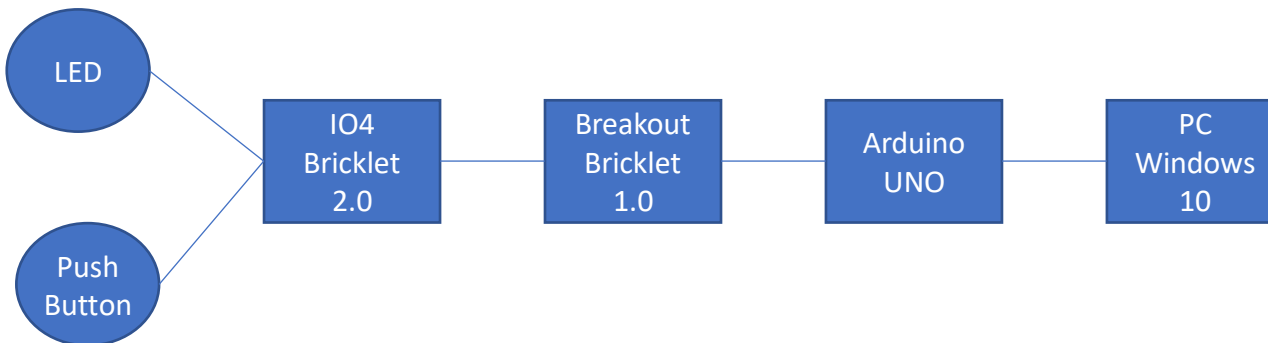
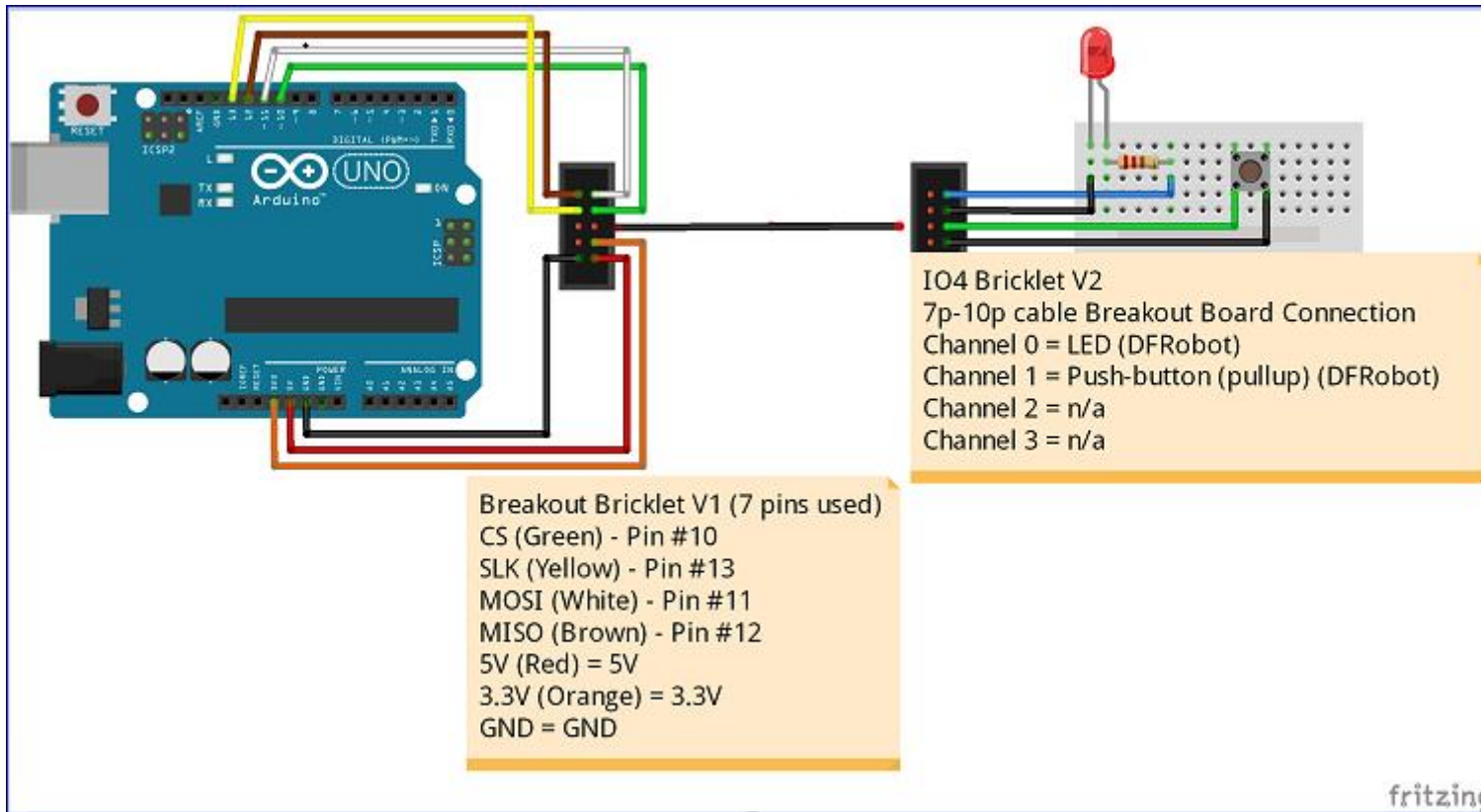
## Software

- Arduino IDE 1.8.13
- Tinkerforge C/C++ Bindings Microcontroller 20200703
- Windows 10 Development Device

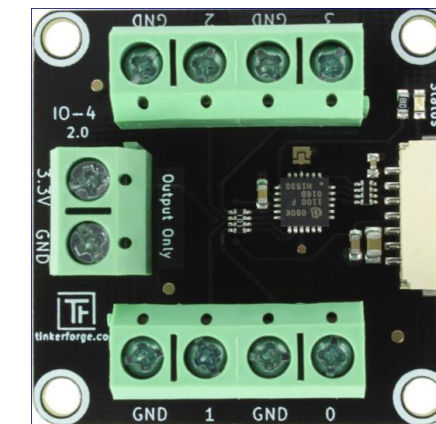
**NOTE:** Check out the source code for latest version.

by R.W.B. Linn 20200707

# Arduino - Circuit



0/AD = CS - SPI Chip Select  
1/DA = SLK - SPI Serial Clock  
2/PWM = MOSI - SPI Master Output, Slave Input  
3 = MISO - SPI Master Input, Slave Output



UID=G4d

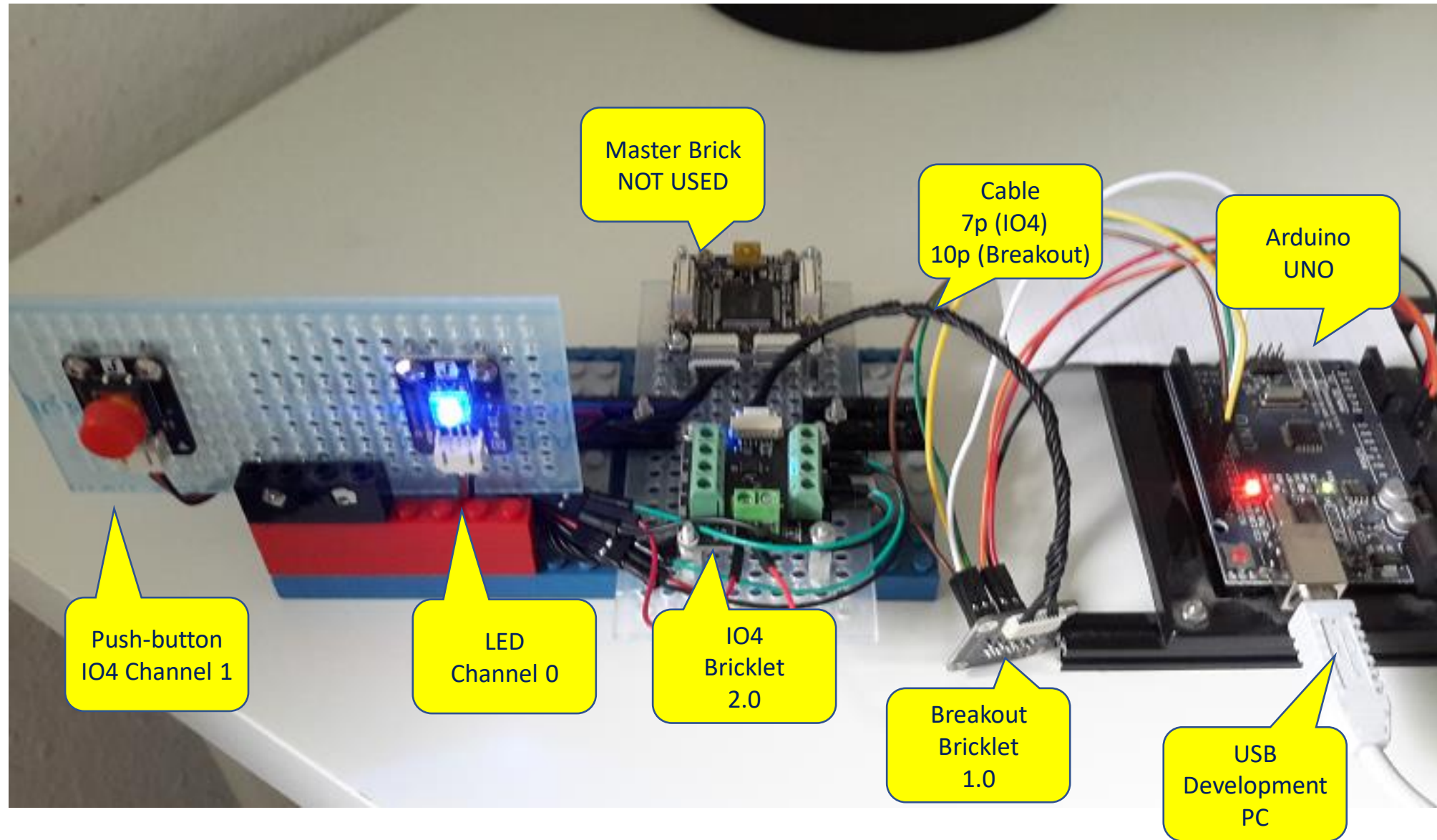
# Arduino – Wiring Breakout Board

SPI	Arduino	Wire Color	Breakout Board
	5V	Red	5V
	GND	Black	GND
	3.3V	Orange	3.3V
	n/a		
	n/a		
	n/a		
	10 – SS	Green	CS – IO1 (0/AD) = SPI Chip Select – set Arduino sketch
	13 – SCK	Yellow	SLK – IO2 (1/DA) = SPI Serial Clock
	11 – MOSI	White	MOSI – IO3 (2/PWM) = SPI Master Output, Slave Input
	12 – MISO	Brown	MISO – IO4 (3) = SPI Master Input, Slave Output

## Notes

- Arduino connected to development device via USB.
- Breakout board connected to IO4 Bricklet 2.0 with 7p (IO4) - 10p (Breakout) cable.
- SPI Chip Select used in the Arduino Sketches.
- Arduino MEGA SPI Pins: 53=SS, 52=SCK, 51=MOSI, 50=MISO.
- Arduino Serial Peripheral Interface SPI [info](#).

# Arduino – Test Setup



# Arduino - Folder Structure

d:\devtools\arduino\experiment\experiment.ino	Arduino sketch experiment
d:\devtools\arduino\experiment\src d:\devtools\arduino\experiment\src\bindings d:\devtools\arduino\experiment\src\bindings\...	Bindings used without changes
d:\devtools\arduino\experiment\src\hal_arduino d:\devtools\arduino\experiment\src\hal_arduino\hal_arduino.cpp d:\devtools\arduino\experiment\src\hal_arduino\hal_arduino.h	HAL used without changes
d:\devtools\arduino\experiment\src\experiment d:\devtools\arduino\experiment\src\experiment\experiment.c d:\devtools\arduino\experiment\src\experiment\experiment.h	Program to control the bricklet(s) of the experiment

## Notes

Replace the “experiment” with the name of the experiment, i.e.

- io4led
- io4ledbtn
- io4ledserial has no src/experiment folder as io4ledserial.ino contains all code

Example for experiment io4led: The files created are io4led.ino, io4led.h, io4led.c

# Arduino - Sketch io4led.ino

io4v2 channels: 0 = output - led (dfrobot)  
logic: the led blinks 10 times with 1s interval.

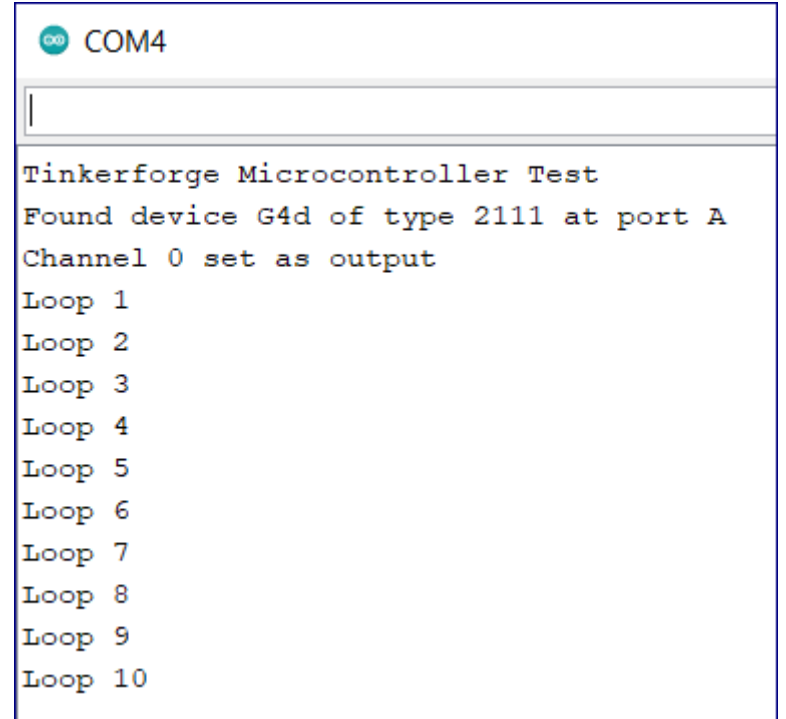
```
#include "src/hal_arduino_avr/hal_arduino_avr.h"
extern "C" void io4led_setup(TF_HalContext *hal);
extern "C" void io4led_loop(TF_HalContext *hal, bool state);

TF_Port ports[1] = {{.chip_select_pin=10,.port_name='A'}};
TF_HalContext hal;
unsigned long previousMillis = 0;
const long blinkInterval = 1000;
unsigned long blinkCount = 0;
const long blinkMaxCount = 10;
bool ledState = false;

void check(int rc, char *msg) {
    if (rc >= 0)
        return;
    Serial.print(millis());
    Serial.print(" Failed to "); Serial.print(msg);
    Serial.print(" rc:"); Serial.println(rc);
    delay(10);
}

void setup() {
    Serial.begin(115200);
    delay(3000);
    Serial.println("Tinkerforge Microcontroller Test");
    check(tf_hal_arduino_init(&hal, ports, sizeof(ports)/sizeof(ports[0])), "hal init");
    io4led_setup(&hal);
}

void loop() {
    unsigned long currentMillis = millis();
    if ((currentMillis - previousMillis >= blinkInterval) && (blinkCount < blinkMaxCount)){
        blinkCount++;
        previousMillis = currentMillis;
        tf_hal_log_info("Loop %d", blinkCount);
        ledState = !ledState;
        io4led_loop(&hal, ledState);
    }
}
```



```
COM4

Tinkerforge Microcontroller Test
Found device G4d of type 2111 at port A
Channel 0 set as output
Loop 1
Loop 2
Loop 3
Loop 4
Loop 5
Loop 6
Loop 7
Loop 8
Loop 9
Loop 10
```

## Notes

- Function loop(): not used delay() as loop() hangs.
- Extended parameter for extern C io4led\_loop() to test handling additional function arguments
- Single port with SPI Chip Select (CS) pin connected to the Arduino UNO pin #10 (MEGA pin #53)

# Arduino - Bricklet Programs io4led.h, io4led.c

```
/*
io4led.h
*/
#include "../bindings/hal_common.h"

void io4led_setup(TF_HalContext *hal);
void io4led_loop(TF_HalContext *hal, bool state);
```

## Notes

- Program io4led.c requires bricklet uid.
- Ensure unique name for the check function to avoid conflicts.

```
/*
io4led.c
*/
#include "io4led.h"
#include <stdio.h>
#include "../bindings/bricklet_io4_v2.h"

TF_IO4V2 io;
UID_IO4V2 = "G4d";

void io4led_check(int rc, char *msg) {
    if (rc >= 0)
        return;
    tf_hal_log_error("Failed to %s: %d", msg, rc);
}

void io4led_setup(TF_HalContext *hal) {
    io4led_check(tf_io4_v2_create(&io, UID_IO4V2, hal), "Create io4");
    // Configure channel 0 as output high
    tf_io4_v2_set_configuration(&io, 0, 'o', true);
    tf_hal_log_info("Channel 0 set as output high");
}

void io4led_loop(TF_HalContext *hal, bool state) {
    tf_io4_v2_set_selected_value(&io, 0, state);
}
```



# Arduino - io4btnled

io4v2 channels: 0 = output - led (dfrobot), 1 = input – push-button (dfrobot)

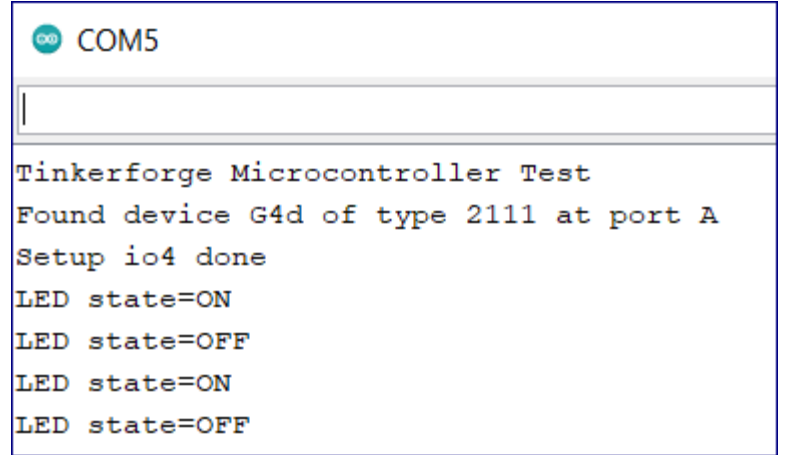
logic: if the pushbutton is pressed the led is turned on or off depending led state.

```
#include "src/hal_arduino/hal_arduino.h"
extern "C" void io4ledbtn_setup(TF_HalContext *hal);
extern "C" void io4ledbtn_loop(TF_HalContext *hal);
// Port A with Arduino CS Pin = UNO #10, MEGA #53
int CSPIN = 53; // 10
TF_Port ports[1] = {{.chip_select_pin=CSPIN,.port_name='A'}};
TF_HalContext hal;

void check(int rc, char *msg) {
    if (rc >= 0)
        return;
    Serial.print(millis()); Serial.print(" Failed to ");
    Serial.print(msg); Serial.print( " rc:"); Serial.println(rc);
    delay(10);
}

void setup() {
    Serial.begin(115200);
    delay(3000);
    Serial.println("Tinkerforge Microcontroller Test");
    check(tf_hal_arduino_init(&hal, ports, sizeof(ports)/sizeof(ports[0])), "hal init");
    io4ledbtn_setup(&hal);
}

void loop() {
    io4ledbtn_loop(&hal);
}
```



```
COM5

Tinkerforge Microcontroller Test
Found device G4d of type 2111 at port A
Setup io4 done
LED state=ON
LED state=OFF
LED state=ON
LED state=OFF
```



# Arduino - io4btnled.h, io4btnled.c

```
/*
io4ledbtn.h
io4v2 channels:
0 = output - led (dfrobot)
1 = input - pushbutton (dfrobot)
logic:
pushbutton pressed, led turned on or off depending led state.
*/

#include "../bindings/hal_common.h"

void io4ledbtn_setup(TF_HalContext *hal);
void io4ledbtn_loop(TF_HalContext *hal);
```

```
#include "io4ledbtn.h"
#include <stdio.h>
#include "../bindings/bricklet_io4_v2.h"
#include "../bindings/errors.h"

TF_IO4V2 io;
IO4_UID = "G4d";
IO4_CALLBACK_PERIOD = 100; // in ms
IO4_TICK_TIMEOUT = 1000; // ms
bool button_pressed = false; // button released
bool led_state = false; // led off

void io4ledbtn_cb_input_value(struct TF_IO4V2 *device, uint8_t channel, bool changed, bool value, void *user_data) {
    (void)user_data; // avoid unused parameter warning
    // If value = true, the button is pressed or released. Init with button released. Inverse led state
    if (value == true) {
        button_pressed = true;
        led_state = !led_state;
    }
}

void io4ledbtn_check(int rc, char *msg) {
    if (rc >= 0)
        return;
    tf_hal_log_error("Failed to %s: %d", msg, rc);
}

void io4ledbtn_setup(TF_HalContext *hal) {
    io4ledbtn_check(tf_io4_v2_create(&io, IO4_UID, hal), "Create io4 device object");
    tf_io4_v2_set_configuration(&io, 0, 'o', false);
    tf_io4_v2_set_configuration(&io, 1, 'i', true);
    tf_io4_v2_register_input_value_callback(&io, io4ledbtn_cb_input_value, NULL);
    tf_io4_v2_set_input_value_callback_configuration(&io, 1, IO4_CALLBACK_PERIOD, true);
}

void io4ledbtn_loop(TF_HalContext *hal) {
    tf_io4_v2_callback_tick(&io, IO4_TICK_TIMEOUT);
    if (button_pressed == true){
        button_pressed = false;
        int rc = tf_io4_v2_set_selected_value(&io, 0, led_state);
        if(rc != TF_E_OK)
            tf_hal_log_error("rc %d", rc);
        tf_hal_log_info("LED state=%s", led_state ? "ON" : "OFF");
    }
}
```

# Arduino – io4ledserial.ino

io4v2 channels: 0 = output - led (dfrobot)

logic: read serial line data. if data = L0, turn led off. if L1, turn on

```
#include "src/hal_arduino/hal_arduino.h"
#include "src/bindings/bricklet_io4_v2.h"
// Define port connected Arduino CS pin: UNO=#10, MEGA=#53
TF_Port ports[1] = {{.chip_select_pin=53, .port_name='A'}};

TF_HalContext hal;
TF_IO4V2 io;
char* IO4_UID = "G4d";
int IO4_CH_LED = 0;
bool led_state;
String serial_data="";

void io4_check(int rc, char *msg) {
    if (rc >= 0)
        return;
    tf_hal_log_error("Failed to %s: %d", msg, rc);
}

void io4_setup() {
    io4_check(tf_io4_v2_create(&io, IO4_UID, &hal), "Create io4 device object");
    tf_io4_v2_set_configuration(&io, IO4_CH_LED, 'o', false);
}

void io4_set_state(int channel, bool state) {
    tf_io4_v2_set_selected_value(&io, channel, state);
}

void check_hal(int rc, char *msg) {
    if (rc >= 0)
        return;
    tf_hal_log_info("%u Failed to %s rc: %d", millis(), msg, rc);
    delay(10);
}

void setup() {
    Serial.begin(115200);
    check_hal(tf_hal_arduino_init(&hal, ports, sizeof(ports)/sizeof(ports[0])), "hal init");
    io4_setup();
}
```

```
void loop() {
    byte ch;
    String cmd_str;

    if (Serial.available()) {
        // Read the serial data character & add to data string
        ch = Serial.read();
        serial_data += (char)ch;
        // Received CR=Command ready
        if (ch=='\r') {
            serial_data.trim();
            serial_data.toUpperCase();
            // Process command
            switch( serial_data.charAt(0) ) {
                case 'L':
                    if (serial_data.length()>1){
                        cmd_str = serial_data.substring(1);
                        if (cmd_str.equals("0"))led_state = false;
                        if (cmd_str.equals("1"))led_state = true;
                        // add more states, i.e. 2 blinking
                    } // length
                    io4_set_state(IO4_CH_LED, led_state);
                    break;
                default:
                    tf_hal_log_info("Serial data not handled: %s", serial_data.c_str());
            } // switch

            serial_data = ""; // Clear the string ready for the next command.
        } // if \r
    } // available
}
```