# Auto-Off Module – www.DuinoKit.com

This module will handle up to 20V and up to 2amps. Measuring under 1 square inch and 8 pins for easy connections. Pins also provide feedback on the power button status. This means the same button you held to turn on your project on, can be pressed again to switch your project off. There are also pins to easily relocate the power on/off button elsewhere on your project. The power is held on by using a digital pinto hold the (+) Hold on the module HIGH. The STATUS pin can be read and then used to shut down your project when pressed if desired, however the STATUS pin is optional and only used if wanting to turn our device off by pressing the button. To power the circuit off, you simply power down a pin (through your code) and the entire power will be cut. This can be done after an event such as a button press or after a specified time interval of time in your project code. The BUTTON pins can be used to place the on/off button separate from the PCB

**Sample Arduino code for use here.**

```
long shutDownTimer = 0;
int shutDownAfterSeconds = 5;
int powerButton = 7;
int powerHold = 10;

void setup() {
Serial.begin(9600); // Open serial communication if needed
pinMode(13, OUTPUT); // initialize pin 13 as output to blink the on-board LED
pinMode(powerButton, INPUT_PULLUP); // Use internal pull-up resistor to hold powerButton HIGH
pinMode(powerHold, OUTPUT);
digitalWrite(powerHold, HIGH); // hold the power mosfet ON
delay(500); // Slight delay before checking power button status
}

void loop() {

if (powerButton == LOW) { // If PowerButton is pressed, Shut Down
shutDown();
}

if (millis() - shutDownTimer < shutDownAfterSeconds * 1000) { // wait for the shut-down timer
shutDown();
}

// shutdownTimer = millis(); // Place this line anywhere you want to extend shut down time


///// Run sketch here //////////

digitalWrite(13, HIGH);
delay(200);
digitalWrite(13, LOW);
delay(200);


Serial.print ("Millis=");
Serial.print (millis());
Serial.print (" shutDownTimer=");
Serial.println (shutDownTimer);
}

void shutDown() {
digitalWrite(powerHold, LOW); // when the time is up then shut it down
delay(1000); // give it time to shut down
}
```