

ANDRÉ FELIPE DUTRA FERREIRA

**ELECTROLUX KNOWLEDGE ASSESSMENT: EMBEDDED ELECTRONIC
SYSTEMS**

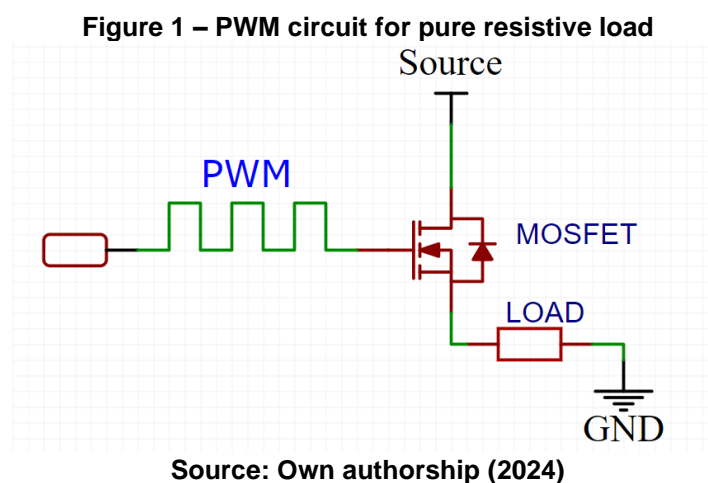
CURITIBA

2024

This document presents some information about the answers for knowledge assessment like images and how to simulate the programs.

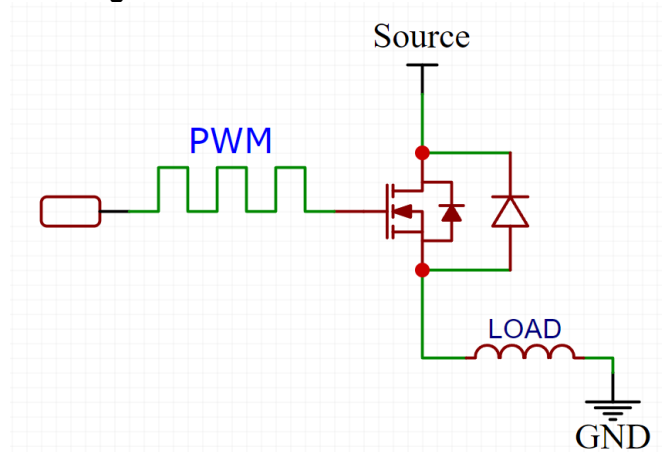
Q1) *Imagine a situation where you have a circuit able to activate and deactivate a purely resistive load with a microcontroller. It is required to control the load power over time depending on the situation and the power cannot be instantly changed from 0% to 100%. Explain in simple words a method to achieve such effect, looking at both hardware and firmware aspects of the system. If an inductive load is used instead of a resistive load, can the same triggering method be used or do special precautions need to be taken?*

A1: For a pure resistive load, it is possible to use a PWM signal. In firmware it can be configured easily. Using a digital pin (with PWM feature) is possible to vary the power supplied to the load, Because the duty cycle percentage of PWM signal corresponds to the percentage of power supplied. For hardware is necessary to separate the Power circuit and the control circuit. To it is possible use a simple circuit presented in the Figure 1.



For an inductive load the Hardware part needs to change a little. Inductive loads create a reverse current when the PWM goes to LOW state. It can damage the MOSFET. To solve it, just add a diode in circuit as presented in Figure 2.

Figure 2 – PWM circuit for inductive load

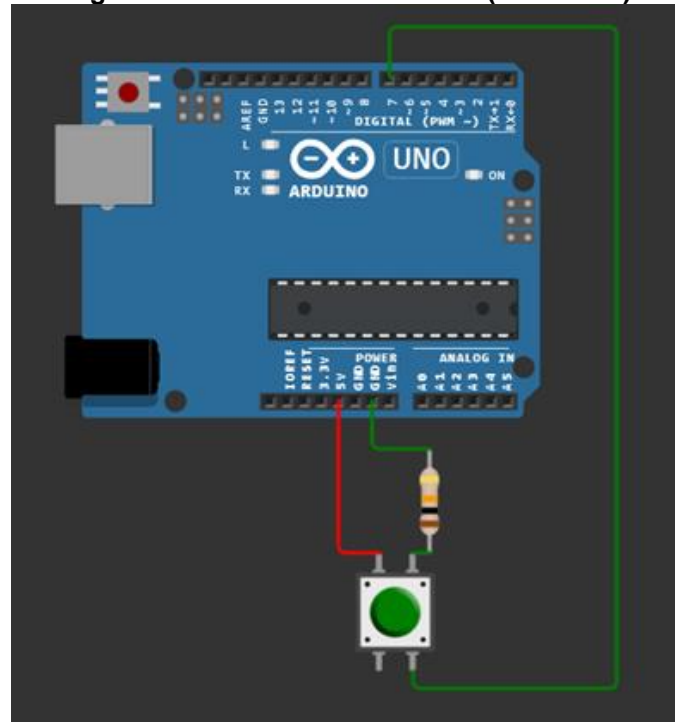


Source: Own authorship (2024)

Q2) *Imagine a situation where you have an electronic board with a microcontroller and a button. You are required to check if the button is pressed (state ON) or not (state OFF) and for how long the button is kept pressed. If the button gets pressed, you also need to have a clear state definition ranging from ON, PROTECTED and OFF; as per the state diagram below. The protected state is defined as an intermediate stage between the transition from the ON state to the OFF state, so that each time the button is released, the button state changes to PROTECTED for 10 seconds before going to the OFF state.*

A2: To this answer look the file “Electrolux_Q2.ino” at [git repository](#). To test the program, it is possible access the link: <https://wokwi.com/projects/new/arduino-uno> and assembly the circuit at Figure 3.

Figure 3 – Arduino button circuit (10 kOhms)



Source: Own authorship (2024)

Q3) *Implement a function able to calculate and return the average, maximum and minimum value of an array with “n” positions. This function must also return a copy of the input array containing only the even numbers of the original array, as well as the new array size.*

A3: To this answer look the file “Electrolux_Q3.c” at [git repository](#).

Q4) *Imagine that you have a microcontroller that communicates to a generic system that may consist of several other boards via UART. How do you ensure that each message is properly sent and that its content is correct? Now imagine that you receive a message, and an interruption is triggered every time a new information is received. Inside every message you have a different command, and for every command you have different payloads, receiving positive and negative values.*

A4: To this answer look the file “Electrolux_Q4.ino” at [git repository](#). To test the program, it is possible access the link: <https://wokwi.com/projects/new/esp32> and assembly the circuit at Figure 4. This program was created to work in ESP32, the circuit can be simulated without the LED, when an interrupt happens the program

writes in serial monitor “event”. During the loop the program writes what was received. The LED indicates the interruption too. When the LED lights off a message was successfully received.

This question cannot be solved using Arduino. It has only two external interrupts (IO 2 and 3). The UART interrupt can’t be configured. The ESP32 board allows configure interrupts for all pins.

Figure 3 – EPS32 LED circuit (1 kOhms)



Source: Own authorship (2024)