Name: Eduardo Betezek Monteiro

Link for Github and the others quastions:: https://github.com/ebm6/ebm\_6.git

Question 1:

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?

One way to control the load power over time is by using a TRIAC circuit. This semiconductor conducts current when it has a positive trigger signal in its gate, that is called the firing angle of the component. Basically, this trigger circuit can be created by using a microcontroller and an optocoupler (to protect the processor) and other resistors for protection or pull-up or pull-down situations depending on each case.

For the firmware side, this circuit should also have a way to detect the zero point of the AC voltage source to synchronize the microcontroller with the supply voltage and then set up the right trigger time for the specific output power that is required for the load and situation. This circuit can be created by using an optocoupler and an AC circuit rectifier to trigger a microcontroller port in which should be configured to be as an interruption one.

After both circuit is created just program the firmware of the microcontroller with the right trigger timings for the right output power it is needed.

For inductive loads, it is needed to implement a snubber circuit in parallel with the TRIAC, this is used to protects it for voltages transients and specially to limit the slope of the reapplied voltage generated by the phase differences from current and voltage in non-resistive loads.

In summary, you can use the TRIAC circuit combined with a microcontroller to fires it, and for an inductive load you should use a snubber circuit for protections and slopes of the reapplied voltages.

Question 4:

Imagine that you have a microcontroller that communicates to a generic system that may consist of several other boards via UART. How do you guarantee each message you send is delivered correctly?

R: CRC.

The other part is at GIThub.