Lab01 Group B4: David Gögl, Ved Varshney, Emre Eryilmaz

Q1 & Q2 It blinks from left to right. Changing the resistor changes the brightness of the LED’s.

Q3 In Part A we first initialize serial communication at 115200 bits per second. Then, we set the pin modes of the 8 pins to OUTPUT. In Part B we first check if there even is an input in our Serial communication inbox. If there is (Which means, we made an input). We read in one byte (8bits) of data from the serial input and write it to the memory address pointed to by (char*)&incoming. After that we convert the incoming to HEX: First we divide by 16 (same as <<4) which give us the most significant bits (The left 4) [stored in dl] as well as he least significant bits [stored in mlt].

Q4 The cast operation ((char*)&incoming) is saving the input at the address where we first set up the value incoming by means of pointers. We also have to cast this “Ausdruck” into the format char as the Serial commant expects. We need to do this as else the Serial input couldn’t read our input. As it parsed the number to char, the output will be the corresponding ASCII Code, followed by a 10 which represents the “line feed character”.

***NOTE: We changed the initialization of the global variable incoming from (int incoming;) to 8uint8_t incoming = 0;) for two reasons: First off, we only need 1 Byte (8 Bits) for this exercise and not the 2Bytes (16Bits) of a normal int. and secondly we assigned the value 0 to the variable to avoid any undefined behaviour.

Q5 They are split up into to nibbles which are saved separately in two variables dl, mlt. By dividing the input “incoming” by 16 (same as >>4) we get the most significant bit (The left part); By applying the modulo operator we get the least significant nibble (The right four).

Q6 8 Bits

Q7 As dl and mlt are representing the upper and the lower 4 bits of data, they are bot 4 bits long. 4 bits gives them a range of values from 0 to 15. Both together they represent the variable incoming wiich we declared of type uint8_t so 8 Bits with which we can represent Data from 0 to 127.

Q8 If we want dl to be 3 and mlt to be 13 we must understand that the value stored in dl in binary is the first 4 digits of our final number and the value stored in mlt in binary is the right part.

3 -> 0011

13 ->1101

0011 1101

If we now arrange these binary numbers, the right way we get 0011 1101 which results in the decimal number: 61.

Setup of the circuit