|  |  |  |  |
| --- | --- | --- | --- |
| Team name: | B7 | | |
| Homework number: | 04 | | |
| Due date: | October 25th | | |
|  |  |  |  |
| Contribution | NO | Partial | Full |
| 1 Massa Giacomo |  |  | *x* |
| 2 Giampà Simone |  |  | *X* |
| 3 Galimberti Claudio |  |  | *x* |
| 4 Micelli Johanna |  |  | *x* |
| 5 Raduzzi Luca Francesco |  |  | *x* |
| Notes: | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Project name | 04B: ADC conversion triggered by timer interrupt with UART | | |
| Not done | Partially done  (major problems) | Partially done  (minor problems) | Successfully completed |
|  |  |  | *x* |
| Explanation:  First, we have configured the ADC (on pin A1) as ADC1\_IN1 and GPIO analog output, as done in class. We set the *timer2 trigger out event* as *external trigger conversion source*. The clock prescaler is divided by 4 and the sampling time is set to the conservative value of 480 cycles. We enabled the ADC1 global interrupt in the NVIC section. In addition, in the ADC settings, we enabled the interrupt event signal at the end of a single conversion.  Then we configured the timer2 at a frequency of 1Hz, by changing the ARR and PSC values. We noticed that the trigger event selection needed to be configured in timer2 as *update event*.  Code:  The ADC interrupt callback tells us when the conversion is done, in order to get the new read value, so as to transmit it via serial communication. We used a binary flag indicating whether the conversion is completed. After getting the new value, we scale it by dividing by the FSR and multiplying by 3.3 V (max measurable tension). In this version, the voltage value is transmitted from the microcontroller to the terminal emulator via serial communication, using the UART\_Transmit() function.  The cycle continues endlessly, reading new values at a rate of 1Hz. After one cycle iteration is done, the flag is reset. | | | |
| Professor comments: | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Project name | 04C: ADC with output on LCD | | |
| Not done | Partially done  (major problems) | Partially done  (minor problems) | Successfully completed |
|  |  |  | *x* |
| In addition to the configurations of the previous version of the project, here we configured the pins PA4, PB1, PB2, and PB15 to PB12 as GPIO output for LCD communication (instead of using serial communication via UART).  Code:  In this version, the microcontroller displays the voltage value directly on the LCD display on the board. The procedure of reading and scaling the voltage values is the same as in the other version.  We printed the voltage value as a float in the first row. In the second row, we printed the progress bar, indicating the voltage on a scale from 0 to 80. We obtained the conversion result by dividing it by the FSR and multiplying it by 80. We cast this value as an integer and then sent it as an argument to the function drawbar();  In the end, we reset the flag and the cycle continues endlessly, reading new values at a rate of 1Hz. | | | |
| Professor comments: | | | |