|  |  |  |  |
| --- | --- | --- | --- |
| Team name: | *B7* | | |
| Homework number: | *HW11* | | |
| Due date: | 20/12/2022 | | |
|  |  |  |  |
| Contribution | NO | Partial | Full |
| 1 *Giampà Simone* |  |  | *x* |
| 2 *Massa Giacomo* |  |  | *x* |
| 3 *Raduzzi Lucafrancesco* |  |  | *x* |
| 4 *Micelli Johanna* |  |  | *x* |
| 5 *Galimberti Claudio* |  |  | *x* |
| Notes: | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Project name | IR communication with UART | | |
| Not done | Partially done  (major problems) | Partially done  (minor problems) | Successfully completed |
|  |  |  | *x* |
| We set TIM2\_CH3 to PWM mode 2 to generate a square wave of frequency 38kHz. We also configured TIM3 to trigger itself with a 2400 Hz frequency, corresponding to the baud rate used for the infrared communication (2400 bits per second).  We connected the IR LED pin PB10 to TIM2\_CH3 and the IR receiver pin PA10 to USART1\_RX. We enabled UART1 in asynchronous mode at 2400 bps and activated its interrupt.  We use the function void transmit\_byte\_IR(char byte): a function that takes a 8 bit char as input, and then starts the transmission. It first sends the start bit 0, then it sends the char bit by bit. It concludes the transmission with the final bit 1.  The transmission frequency is regulated by a flag updated in HAL\_TIM\_PeriodElapsedCallback.  The transmission of a bit via IR at the pre-defined baud rate requires the flag = 0. The PWM is started for transmitting a 0, and stopped when transmitting a 1. The flag is reset to 1 after completing the transmission of the bit.  Another function void transmit\_string\_IR(char[] string) is used to transmit an entire string, character by character, using the previously mentioned function.  Using the HAL\_UART\_RxCpltCallback() the string is received on the UART1 interface, and saved in a variable and is transmitted using HAL\_UART\_Transmit(&huart2) to the emulated terminal. | | | |
| Professor comments: | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Project name | IR communication with Matrices of buttons and leds | | |
| Not done | Partially done  (major problems) | Partially done  (minor problems) | Successfully completed |
|  |  |  | *x* |
| This project is based on IR communication and our previous project on the keyboard. Pressing a button will send its corresponding alpha-numerical character via the UART interface. Then the matrix of LEDs will display that exact character after receiving it via UART.  We set the IR\_LED pin, IR\_RX pin, UART1, TIM3, and TIM2\_CH3 PWM exactly like in the previous project; additionally, we configured the necessary pins for the keyboard, we set the SPI pins and we enabled SPI1 and his relative DMA\_TX in normal mode. We set TIM10 to trigger itself every 4 ms and enabled its interrupt, and is used for iterating through the matrix-led columns  The transmission via IR is handled in the same way as the previous project.  In the main loop, we handle the button presses. If a button press is detected we call transmit\_byte\_IR(character) to send the corresponding character via the UART interface. We also keep track of the button presses, so that a button press is sent only once and not repeated over time.  In the HAL\_UART\_RxCpltCallback, we handle the reception of the data and save the received character. When TIM10 triggers, we call HAL\_SPI\_Transmit\_DMA() to change the enabled column of the led matrix.  For every character in the keyboard, we associate a structure containing the coordinates of the LEDs that when illuminated, display the corresponding symbol. | | | |
| Professor comments: | | | |