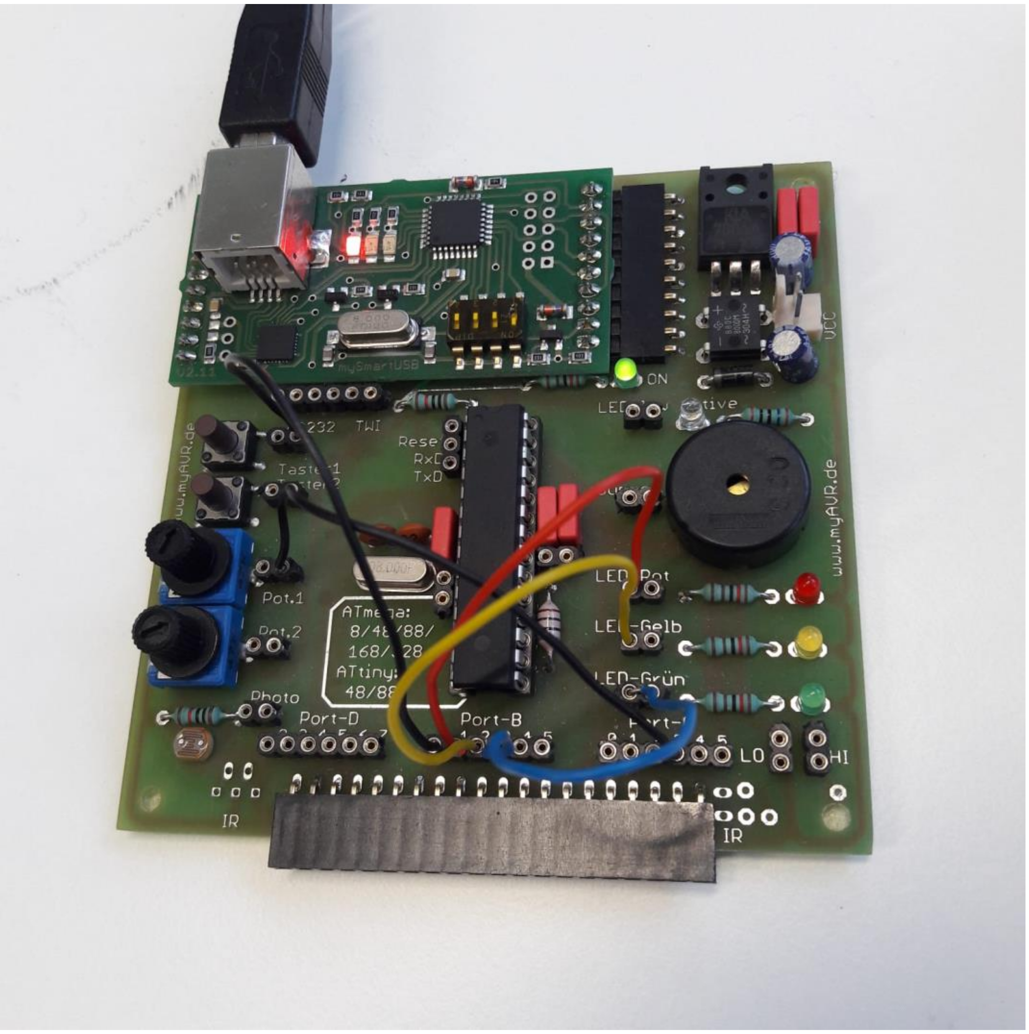
MicroControllers

Lab2

William L. Cole & Romeo Djeulong



Circuitry

Key 1 connected to PB0

Red LED connected to PB1

Yellow LED connected to PB2

Green LED connected to PB3

Task 4: Pot 1 connected to PC3

Task 5: Light Sensor connected to PC3

Programs

Task 1: A while-loop is used to see if Key 1 is pressed. If it is, we use an |= operator to turn the Red LED on by activating PB1.

Task 2: In order to debounce the button, and allow it to toggle LED on or off with only one push

- The program starts with an if-statement, checking if the button is pushed.
- If the button is pushed, there is a short delay (100 milliseconds). This allows the signal time to settle to a steady state behavior.
- Still within the if-statement, the program subsequently uses a bitwise exclusive or modifier '^=' to check the state of the LED and toggle it to the opposite state.
- The if-statement finishes with an empty while loop and another short delay, in order to debounce the output.

Task 3: In order to momentarily dim the LED through Pulse Width Modulation with a sustained push of the button

- The program starts with an if-statement to check if the button is pushed.
- If the button is pushed the LED is immediately turned off with the '&=' operator.
- There is a subsequent delay of 50 milliseconds. This delay sets the period of the 'off' pulse.
- Following the delay the LED is turned back on again with the '|=' operator.
- Finally there is another 50 millisecond delay, setting the period of the 'on' pulse.
- The if-statement is followed by an else-statement (for if the button isn't pressed). Within this else-statement, the LED is simply turned on, meaning that it will be at full 'brightness' when the button is not pressed.

Task 4: In order to convert an analog signal from Potentiometer 1, the program opens by declaring an unsigned 16 bit integer called “converted_value”. The ADC is 10 bits, so the 5Volt signal is divided into a resolution of 1024. Therefore boundary values are:

$$1.66V * 1024/5V = 340$$

$$3.32V * 1024/5V = 680$$

Each of the LED’s needs its own if/else-statements, checking to see if converted_value is within the specified range for that LED. If it is, the LED is turned on, otherwise the LED is turned off.

- The Red LED on PB1 only needs to check if converted_value is greater than or equal to 680
- The Green LED on PB3 only needs to check if converted_value is less than or equal to 340
- The Yellow LED on PB2 must check that converted_value is both greater than 340 and less than 680.

The program finishes with a 1 millisecond delay, to ensure maximum responsiveness.

Task 5: This task uses the exact same programming as Task 4, but ADC3 (PC3) is connected to the Light sensor instead of Potentiometer 1.