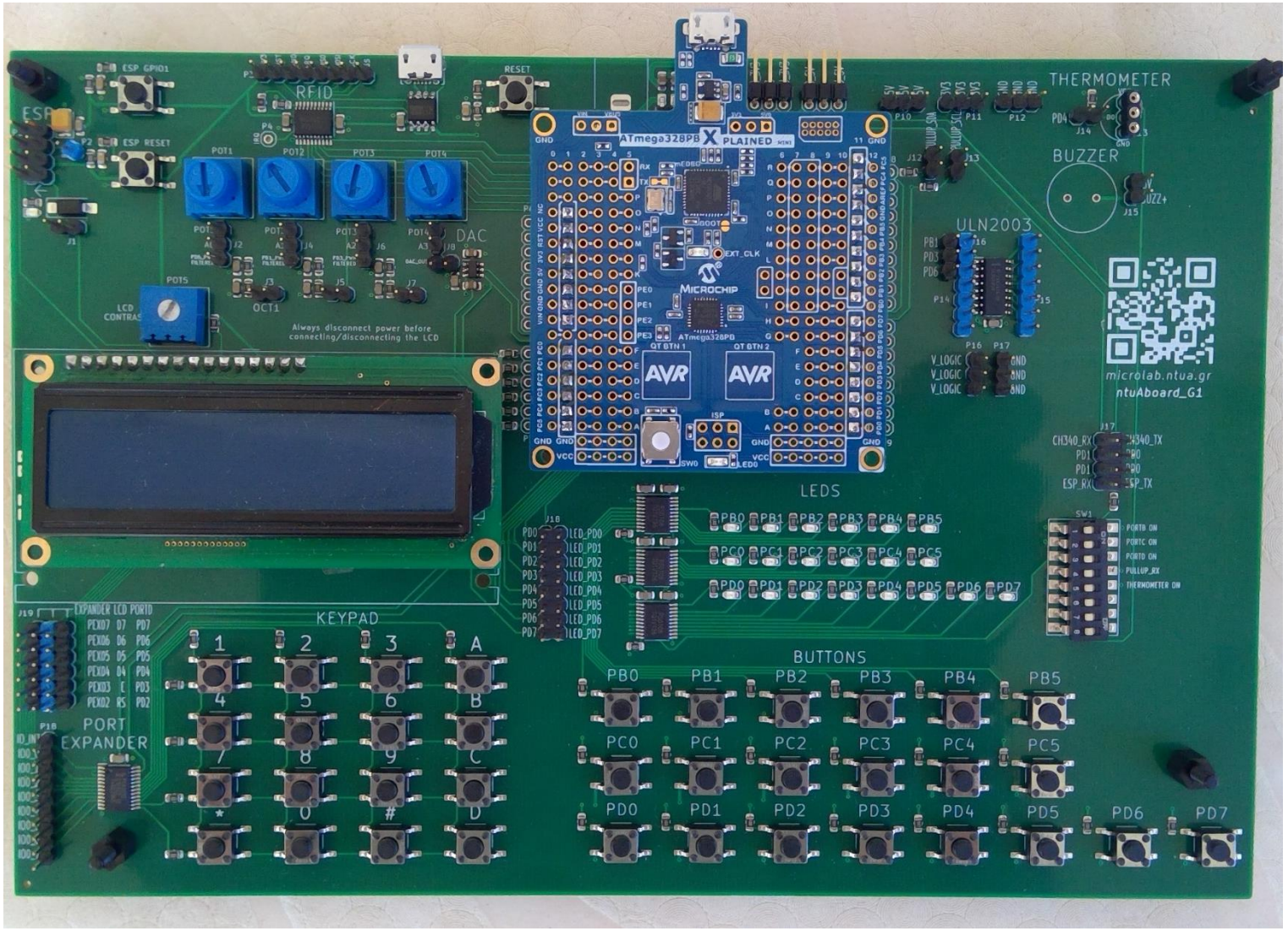
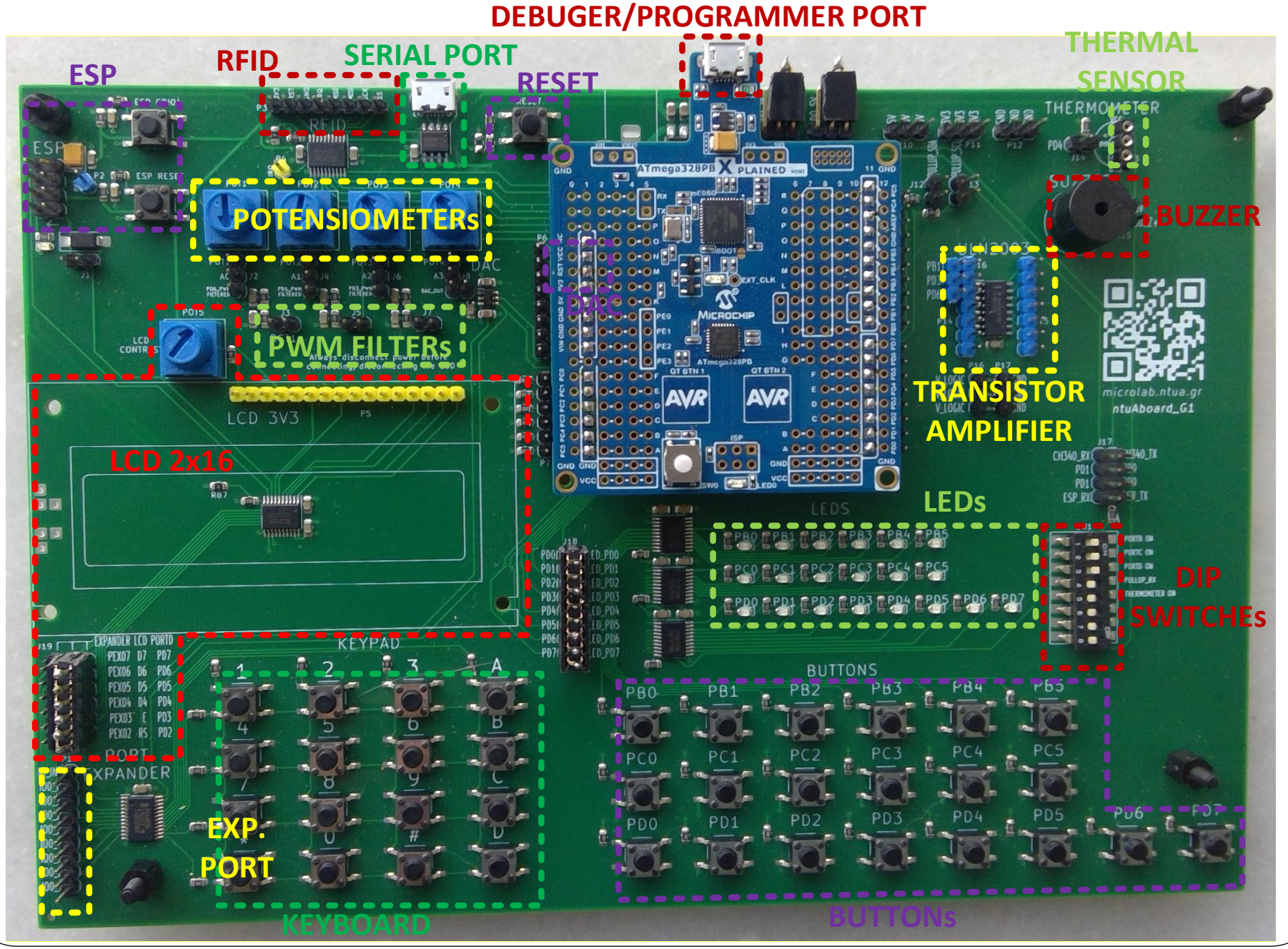
USER GUIDE of ntuAboard_G1 card

National Technical University of Athens
School of Electrical and Computer Engineering
Department of Information Technology and Computers
Microprocessors and Digital Systems Lab

ntuAboard_G1



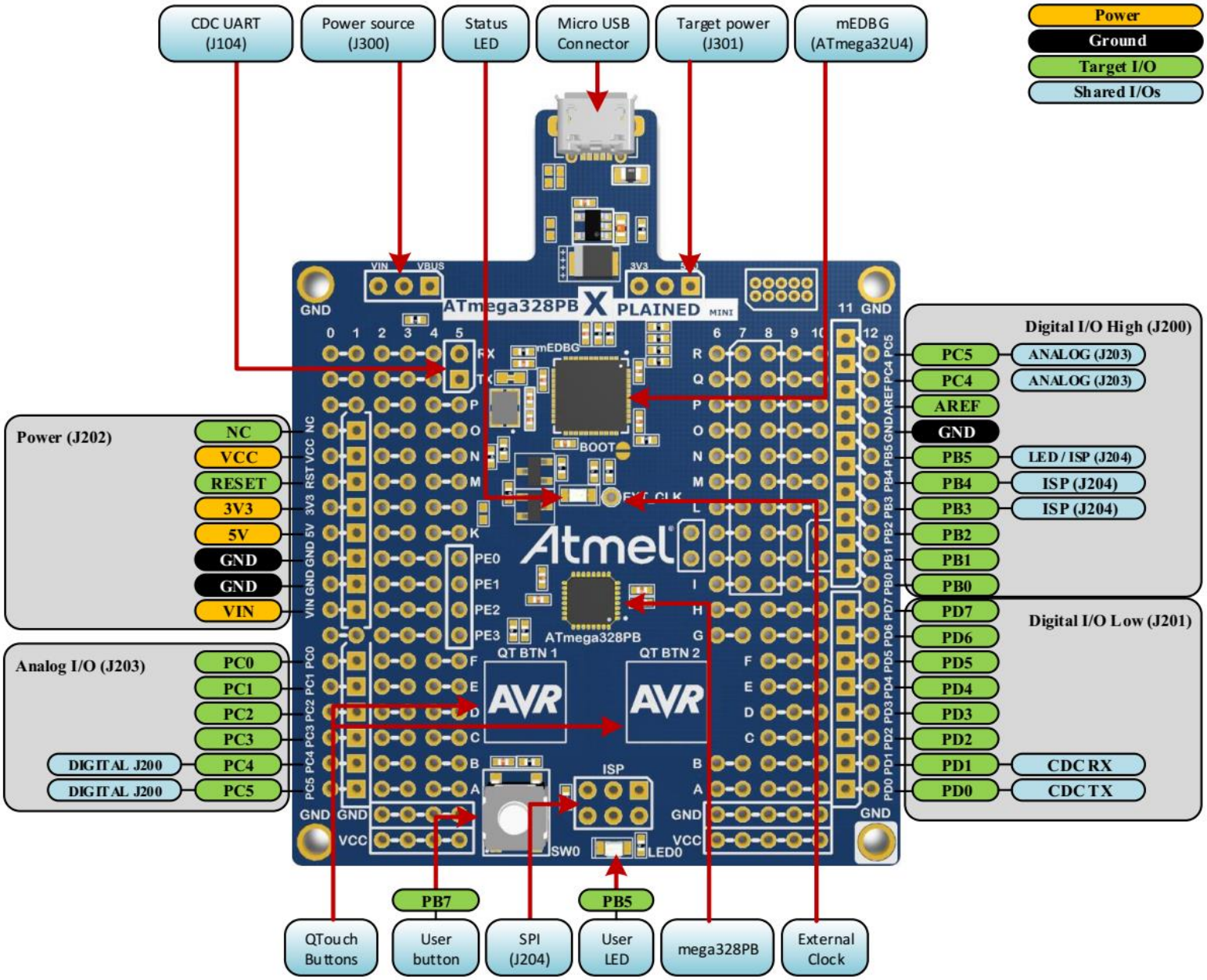
ntuAboard_G1



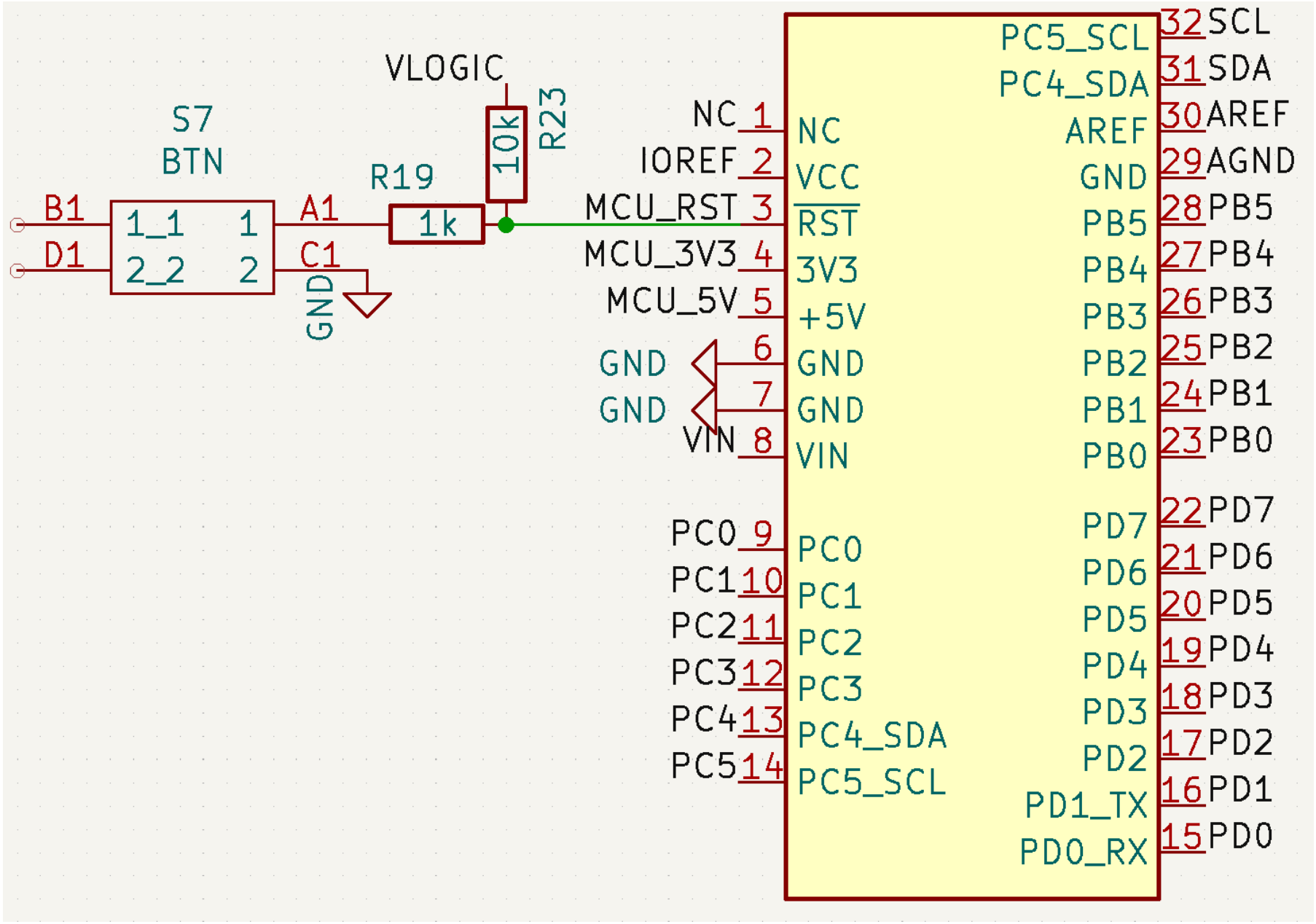
Brief description ntuAboard_G1

- The ntuAboard_G1 has interface contacts compatible with Arduino shield and can accept microcontrollers of various technologies. The main peripheral devices it contains are:
- Indicative Leds.
- Buttons.
- 16-pin I/O expansion port with I2C connection.
- LCD display 2 x 16 characters.
- 4 X 4 Keyboard (16 keys).
- Buzzer for sound production.
- 4 Potentiometers with adjustable analog voltages.
- Digital to analog converter (DAC) with I2C connection.
- 3 analog filters for generating analog voltages from PWM waveforms.
- 7 Darlington Transistor Drivers for driving motors, Power Leds, Buzzers etc.
- Temperature sensor.
- USART to USB serial port adapter for PC connection.
- Interface socket of the ESP-01 WiFi Module.
- Interface socket of the RC522 RFID Development Kit.
- I2C, SPI, USART serial port interface connectors.

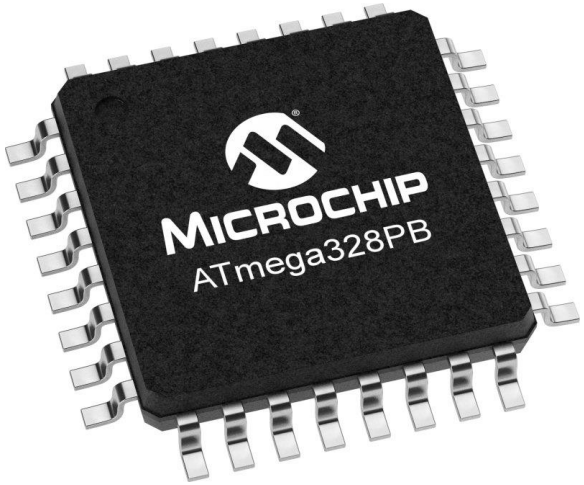
ATmega328PB Xplained Mini Headers and Connectors



MicrochipXplainedMini Pinout



Microcontroller ATmega328PB



- The ATmega328PB is used in many commercial applications.
- It is used in Arduino UNO.
- Information at:

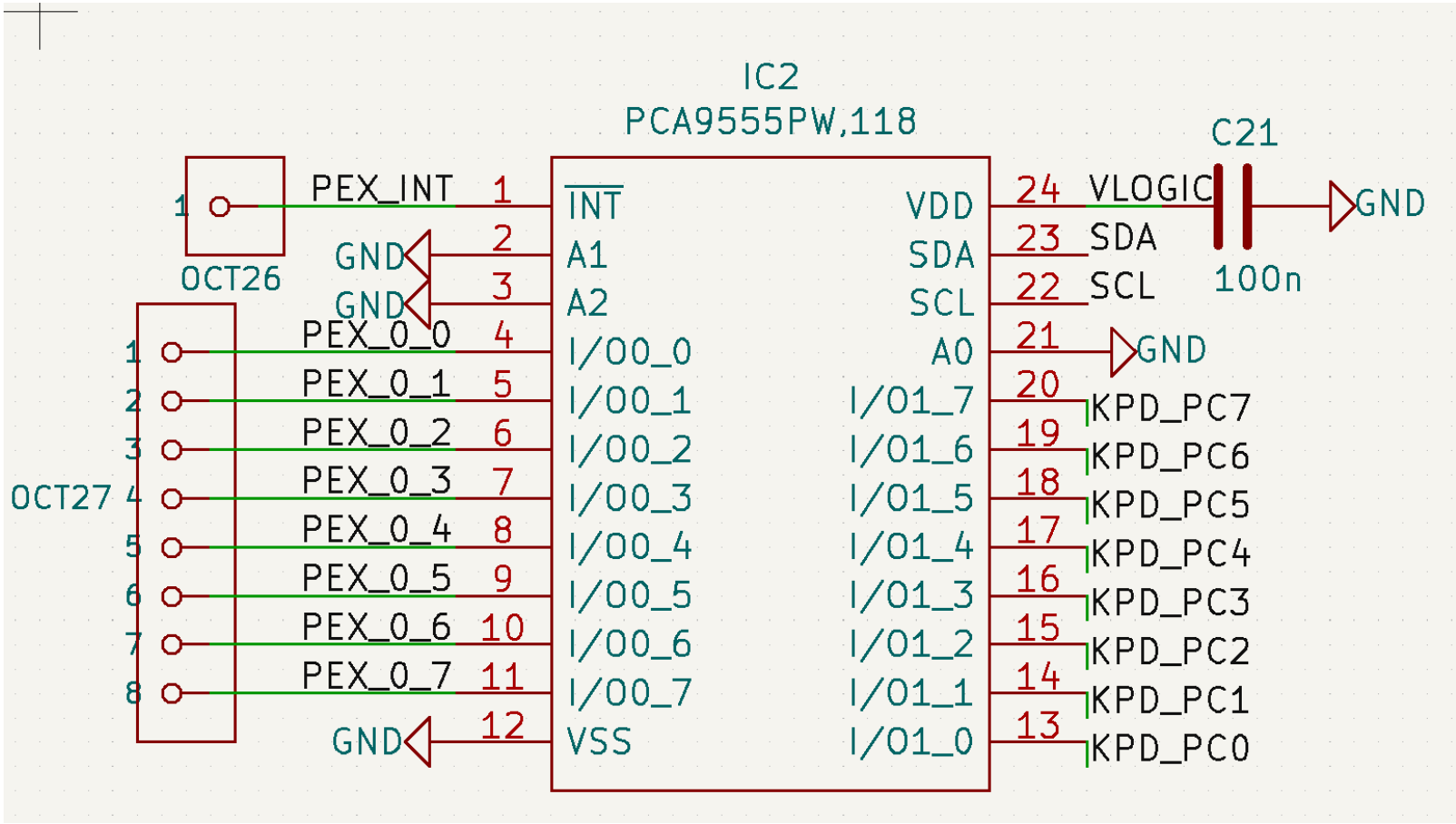
https://www.microchip.com/en-us/product/ATmega328PB

Basic features of ATmega328PB Xplained Mini

- Built-in mini programmer/debugger (mEDBG).
- Auto-ID to detect the card from MPLAB® X.
- Access to all pins of the ATmega328PB microcontroller.
- A green LED indicating mEDBG status.
- A general purpose yellow LED.
- A general purpose push button (Push Button).
- Two touch sensors (QTouch®).
- Virtual COM Port (Virtual COM Port - CDC).
- Built-in crystal 16 MHz(5V) or 8 MHz(3.3V).
- Powered via the USB port.3.3V voltage regulator.
- Compatible Pins for Arduino Shields.
- Contacts for SPI Bus Header.

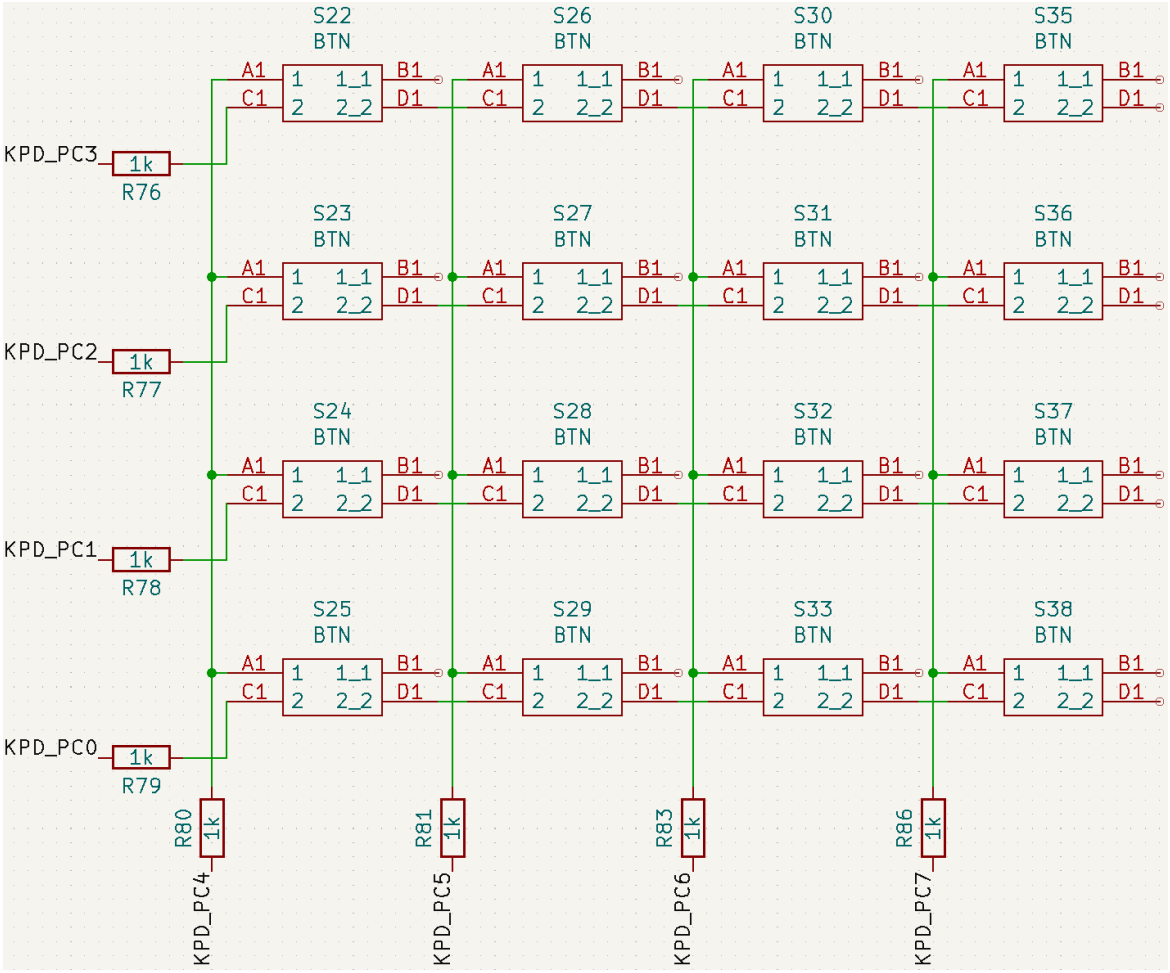
PORT EXPANDER

- It is connected to the microcontroller through the I2C interface (2 pins).
- Produces two I/O ports of 8 bits each.



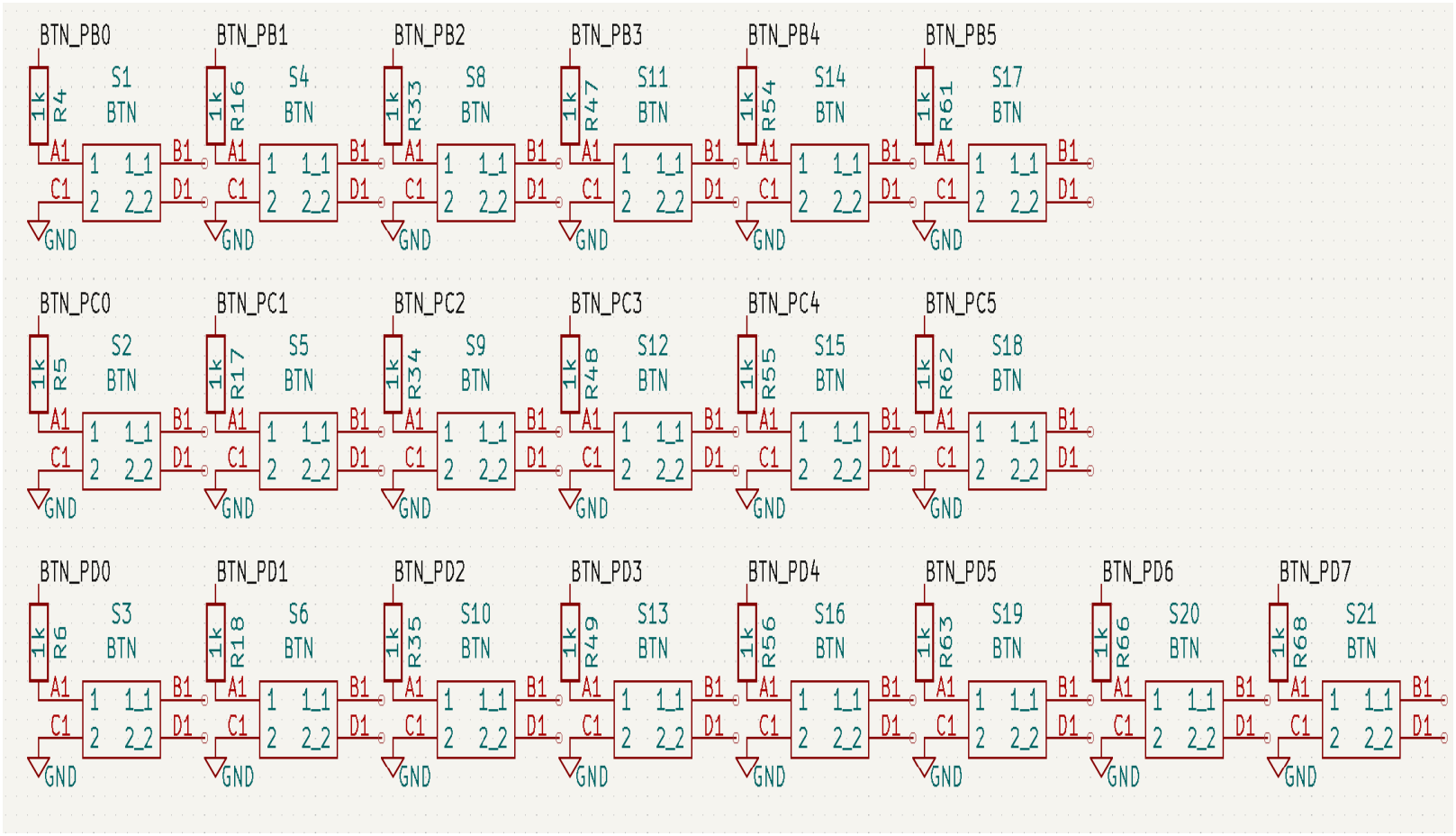
KEYPAD

- It is a 16-key keyboard.
- It is connected to one of the two ports of the PORT EXPANDER.
- Its wiring is shown in the figure below:



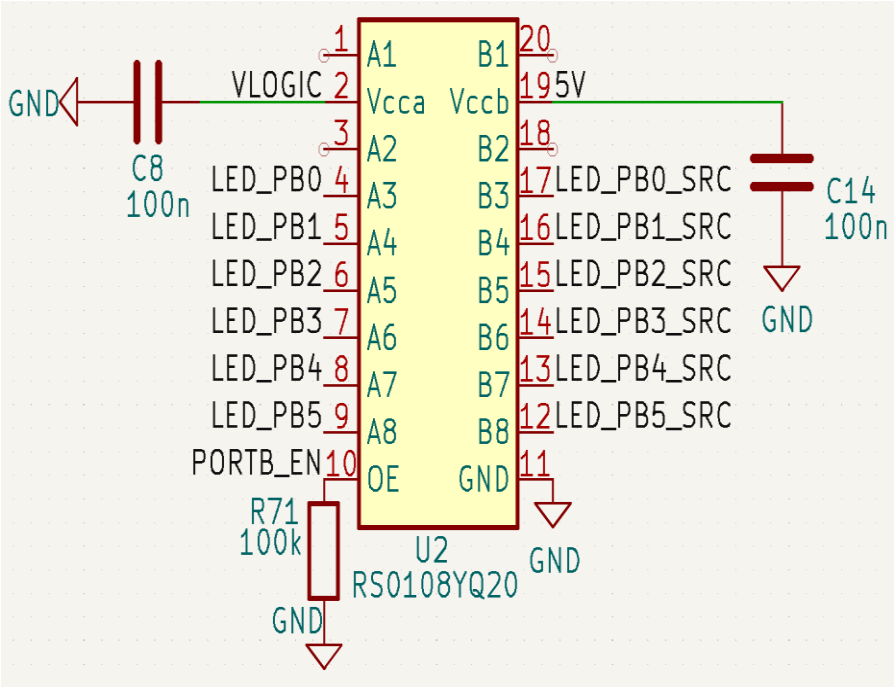
BUTTONS

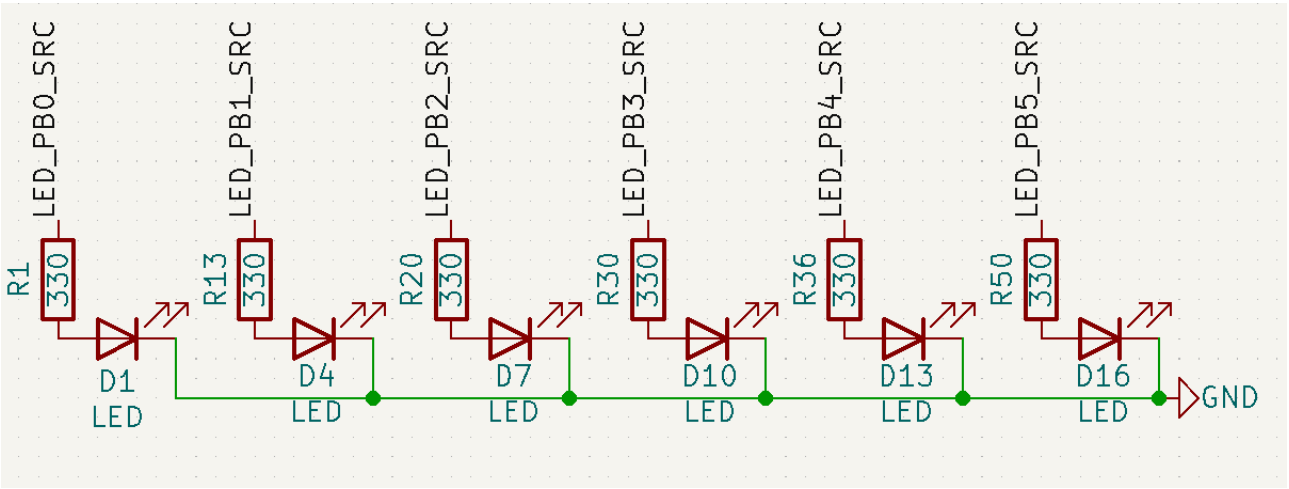
- There is a button for each pin of the microcontroller.
- One pin of each button is connected to GND and the other pin is connected to a pin of the microcontroller.
- A resistor is inserted in each terminal to protect against high current on the microcontroller pins.
- Its wiring is shown in the figure below:



PORTB LEDS

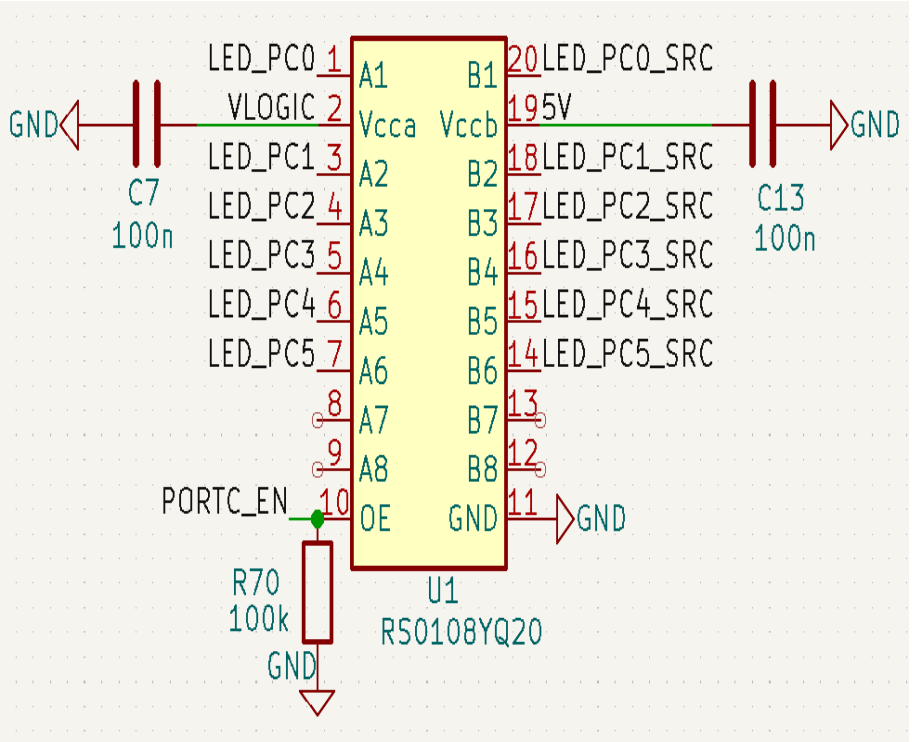
- A led is connected to each terminal of PORTB.
- A chip(IC) is inserted between PORTB and the leds to adjust the microcontroller voltage and to isolate the leds from PORTB when required.

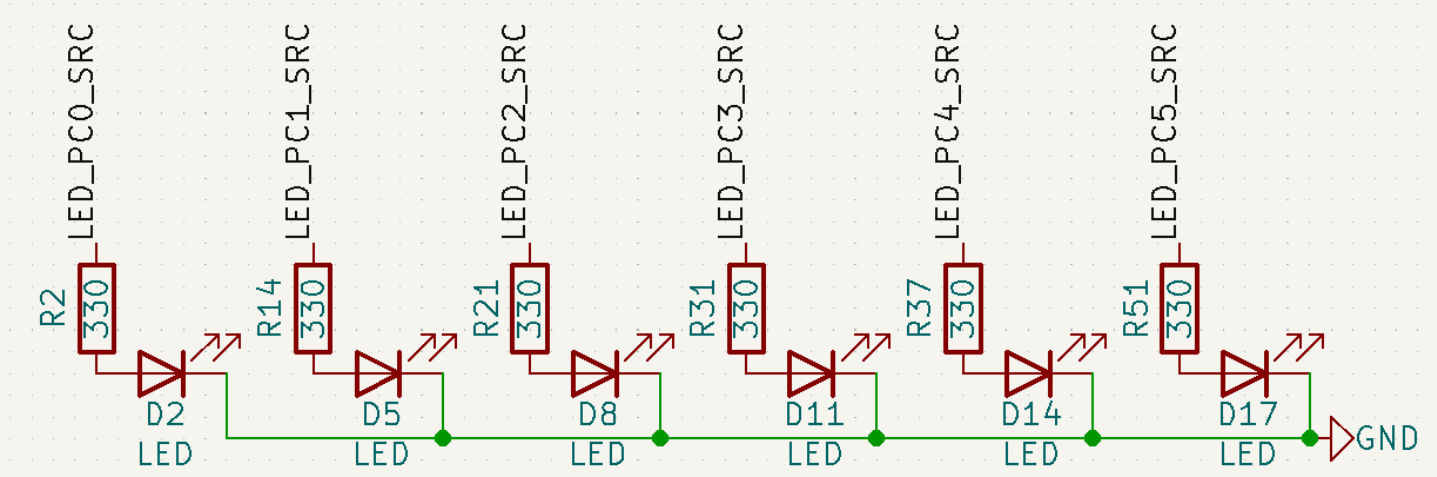




PORTC LEDS

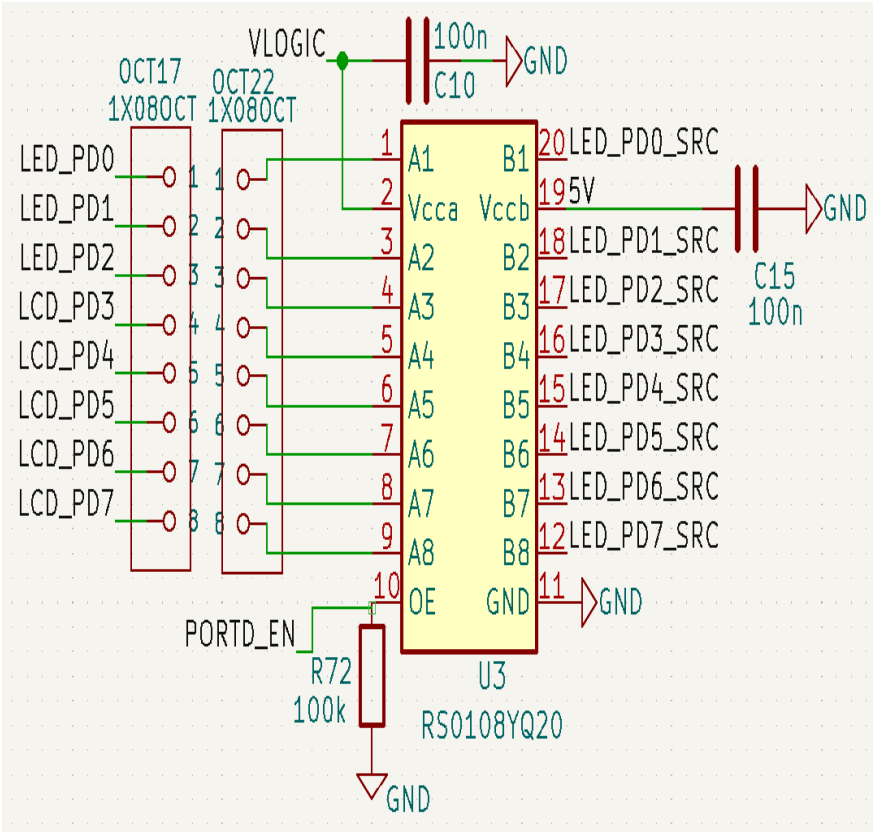
- A led is connected to each terminal of PORTC.
- A chip(IC) is inserted between the PORTC and the leds to adjust the microcontroller voltage and to isolate the leds from the PORTC when required.

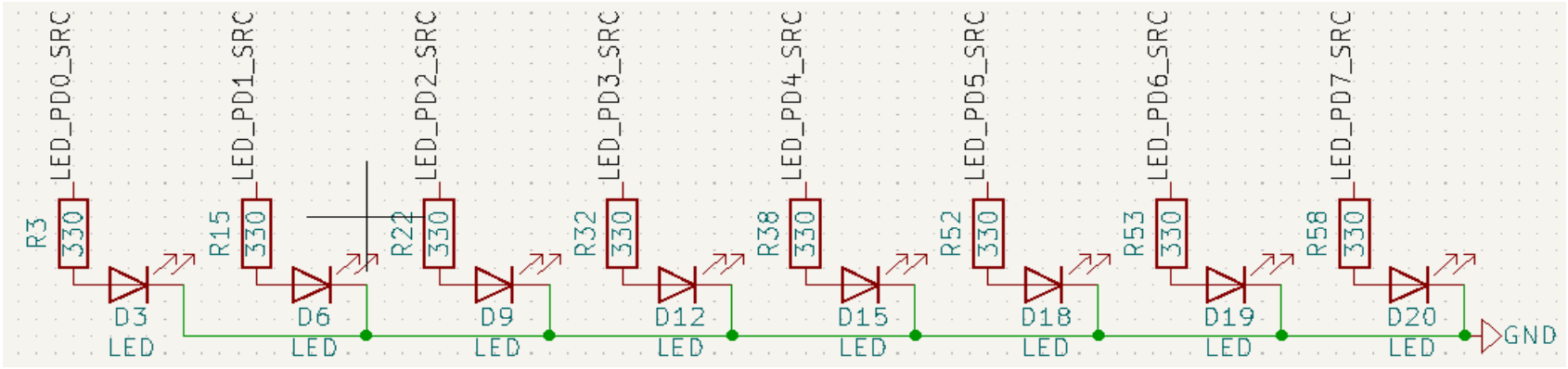




PORTD LEDS

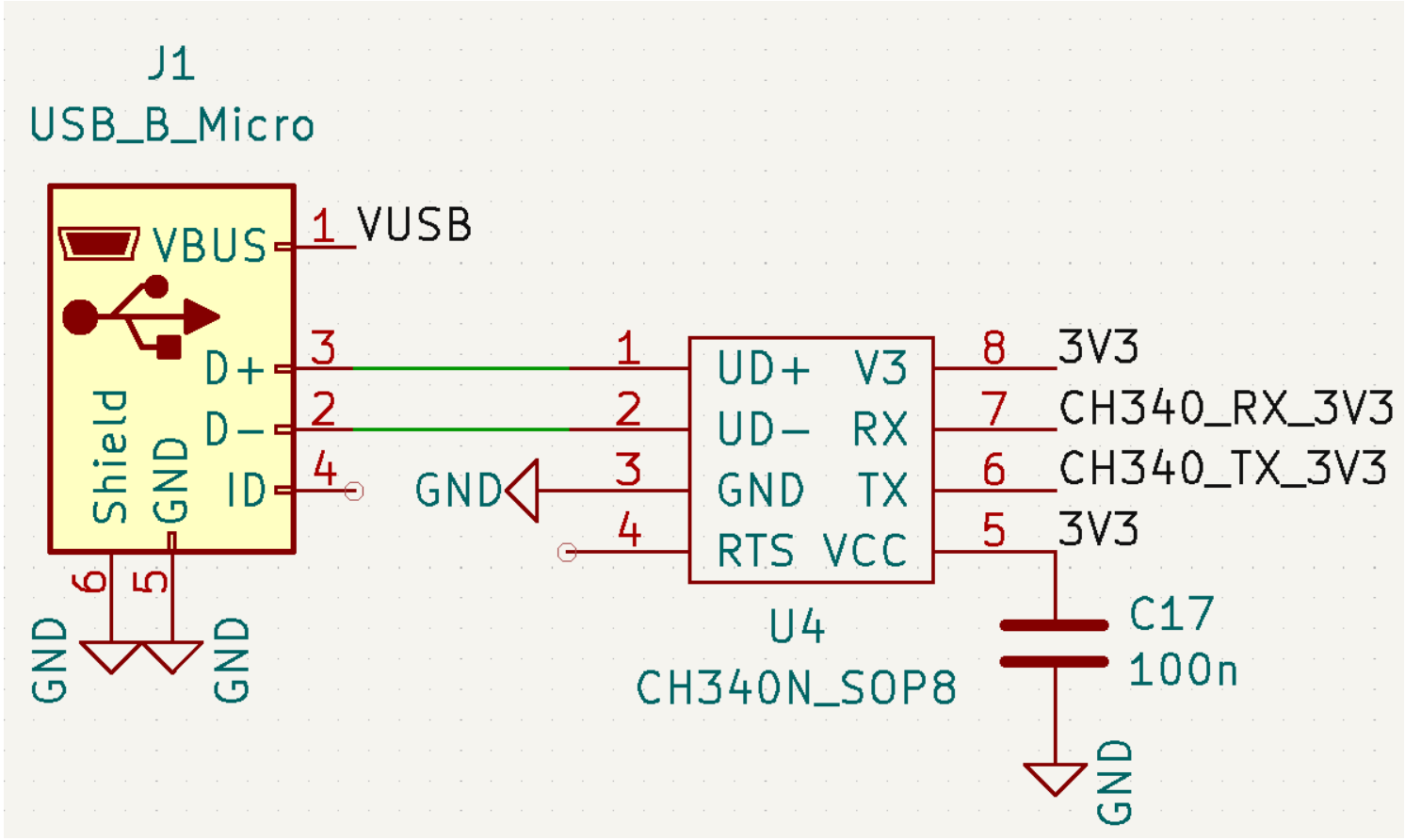
- A led can be connected to each terminal of PORTD.
- A chip(IC) is inserted between PORTD and the leds to adjust the voltage and to isolate the leds when required.
- The leds are connected to the OCT22 connector. PORTD is connected to the OCT17 connector, so by using jumpers the necessary connections are made.
- Alternatively, the terminals of the extended port can be connected to these LEDs with a cable.





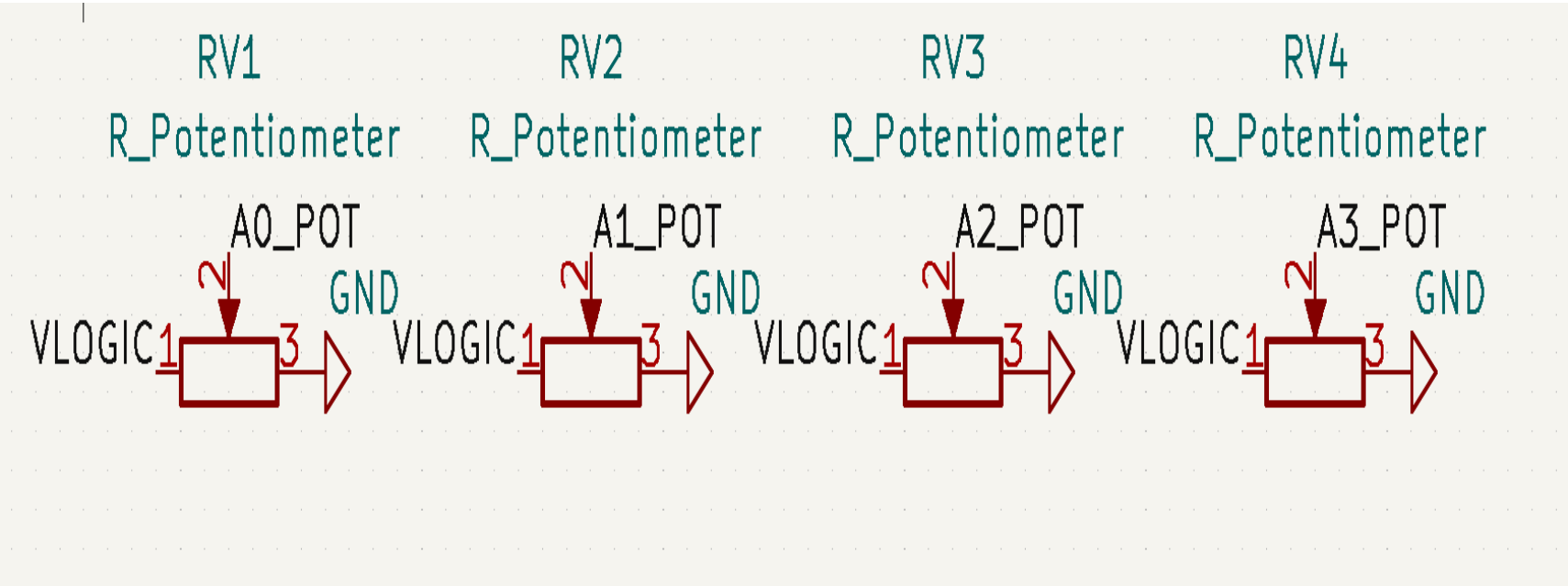
USB TO SERIAL

- This device is used to connect the serial Port of the microcontroller to a USB port of a PC.



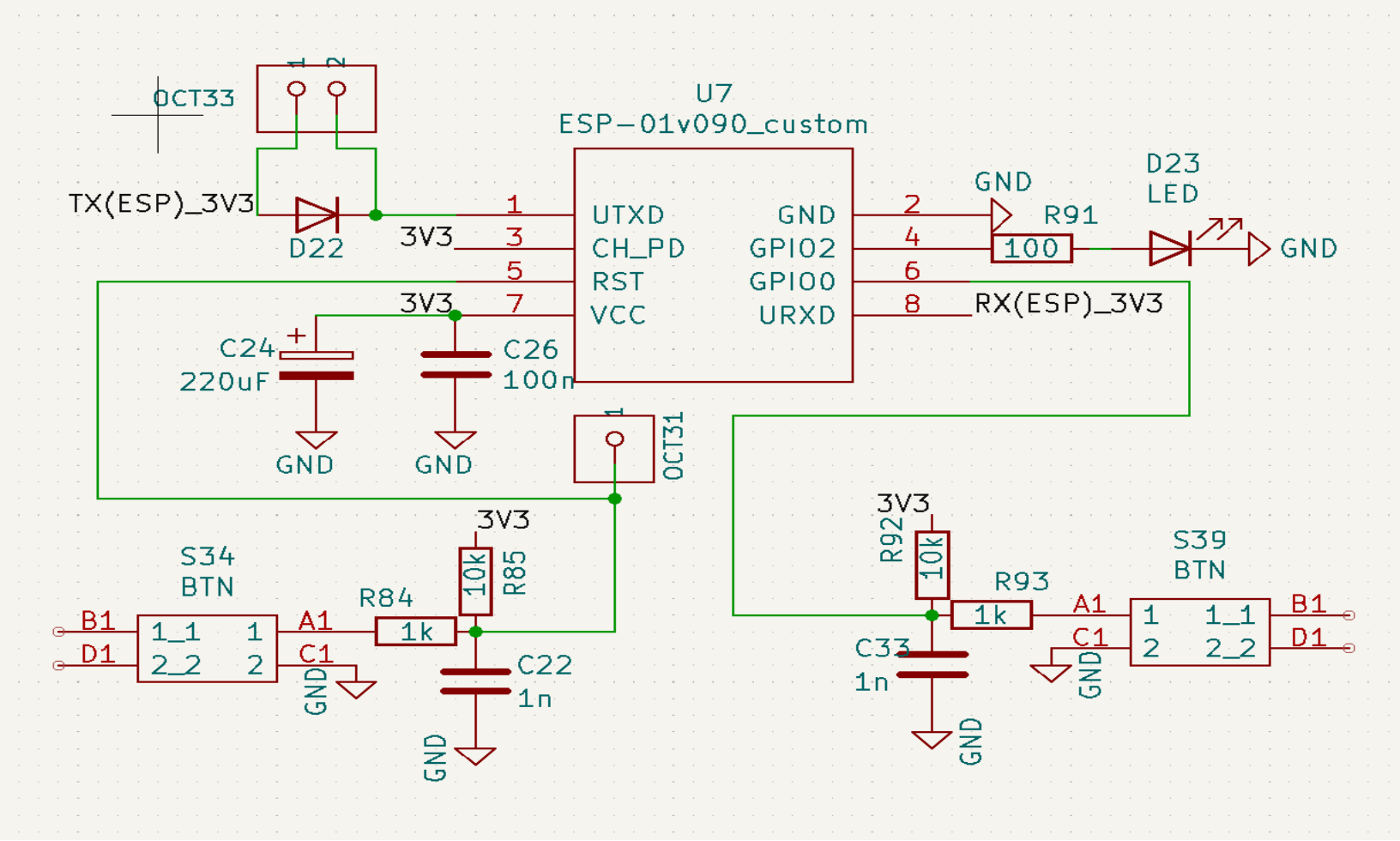
POTENSIOMETERS

- ntuAboard_G1 has 4 potentiometers to produce 4 analog voltages, which can be used to control the ADC converter.



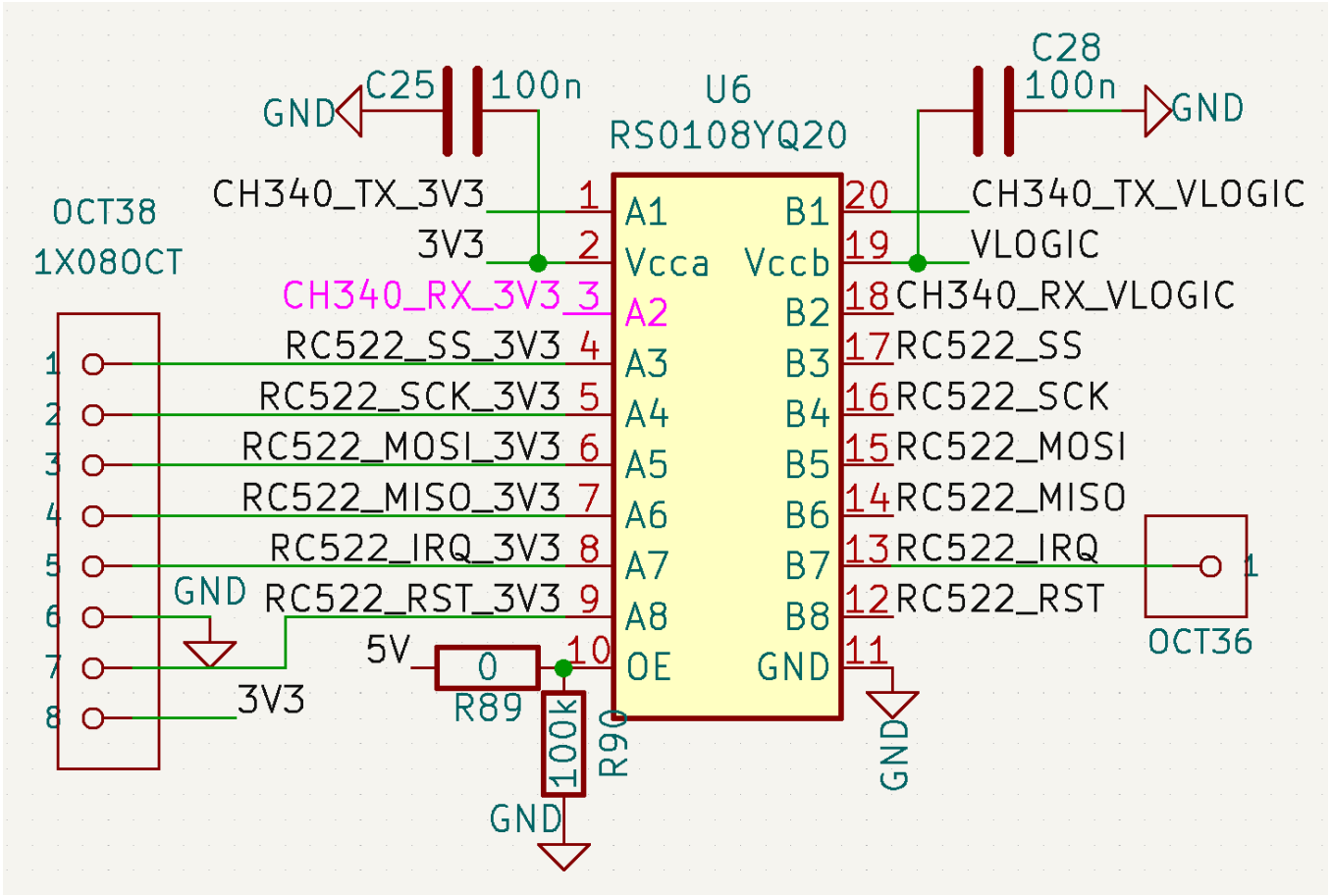
ESP8266 WiFi Module

- The ntuAboard_G1 has one 8-pin connector (P1) for ESP8266 WiFi Module. This module can establish wireless communication.



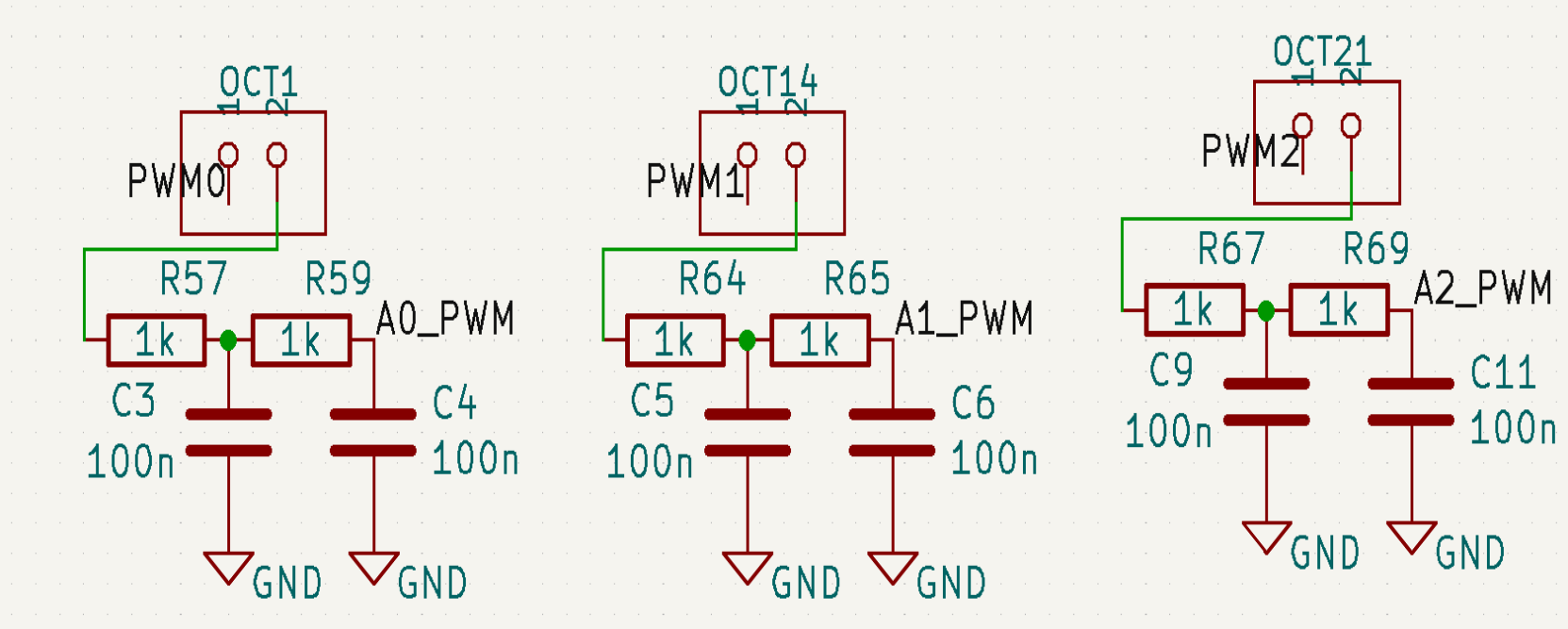
RFID

- The ntuAboard_G1 has one 8-pin connector (OCT38) for RC522 RFID Development Kit. An integrated is inserted between the microcontroller and the RC522 RFID for voltage adjustment and isolation when required.



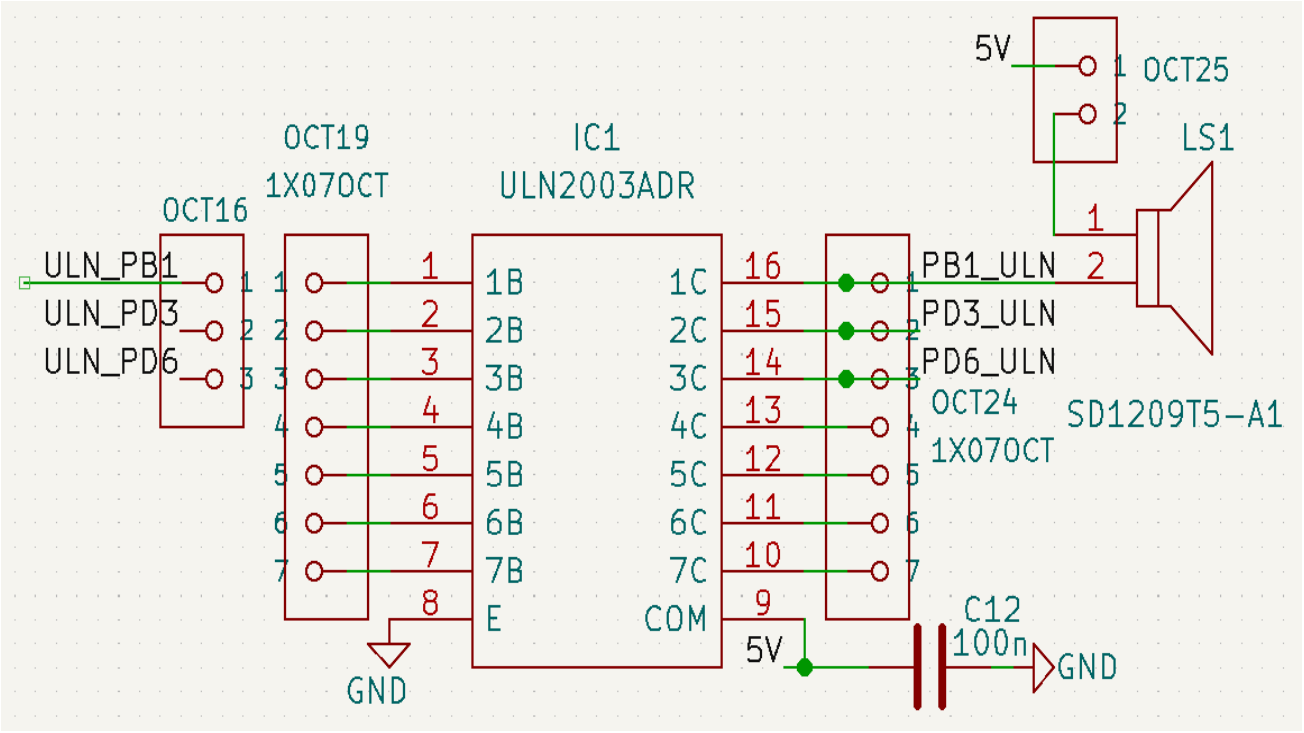
PWM FILTERS

- The microcontroller can generate PWM waveforms with a variable duty cycle.
- The ntuAboard_G1 has 3 analog filters which are used to produce variable DC voltages from the PWM outputs.



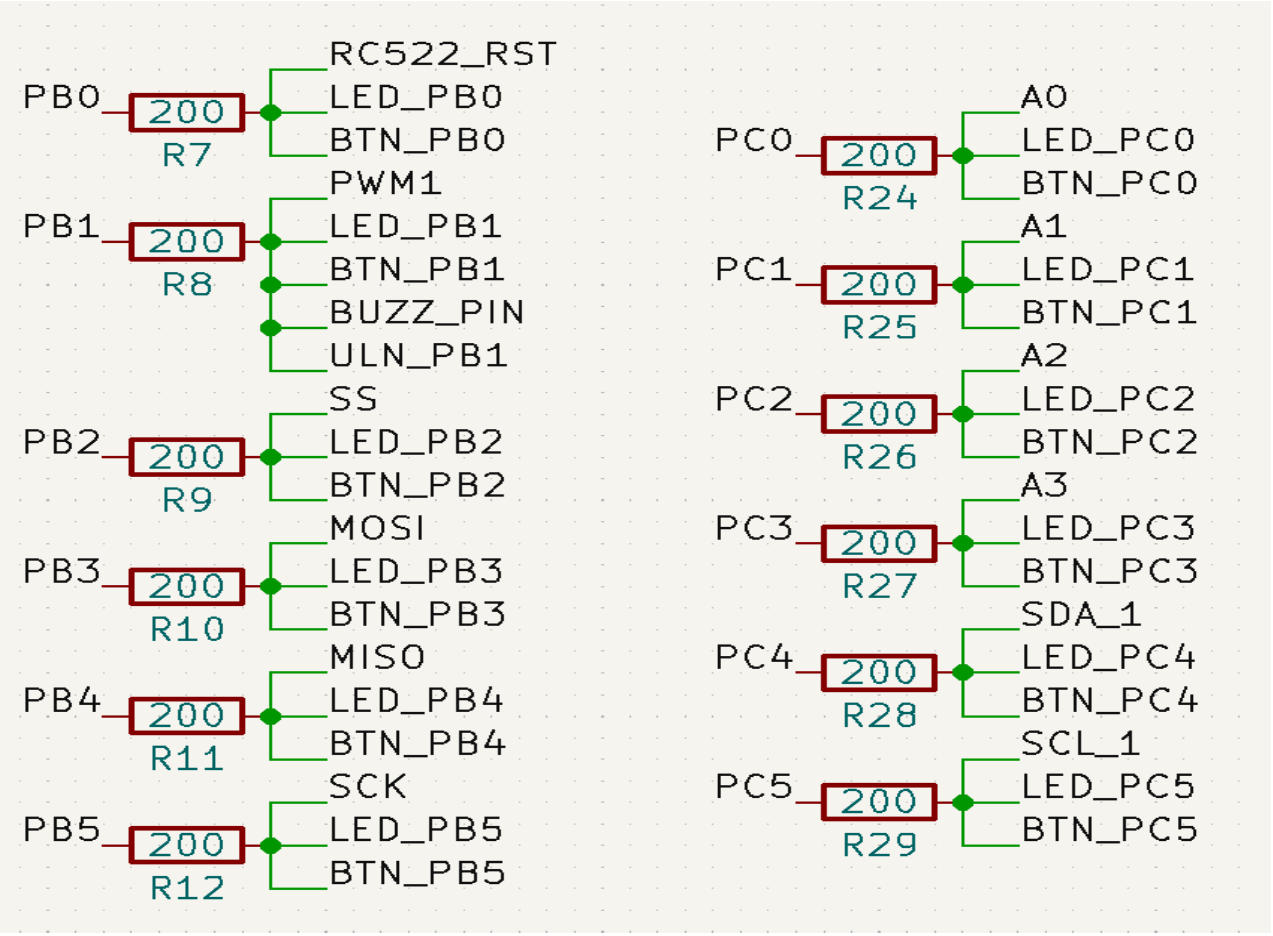
BUZZER and TRANSISTOR ARRAY IC

- The ntuAboard_G1 has a built-in IC of a 7-channel amplifier of darlington transistors.
- A buzzer can be connected to one of these channels.



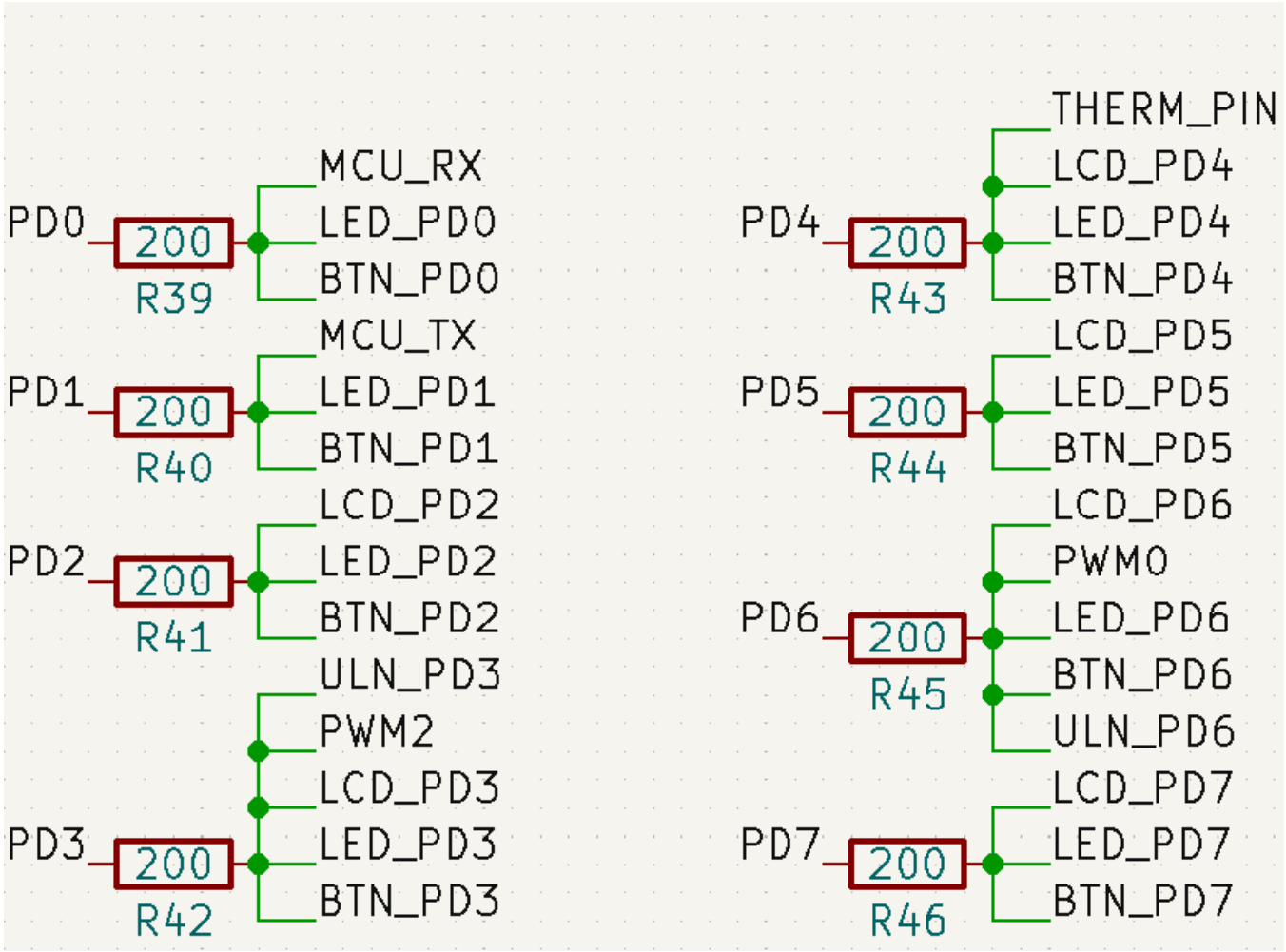
PORTB and PORTC CONNECTIONS

- The figure below shows the various connections available for each of the PORTB and PORTC pins.



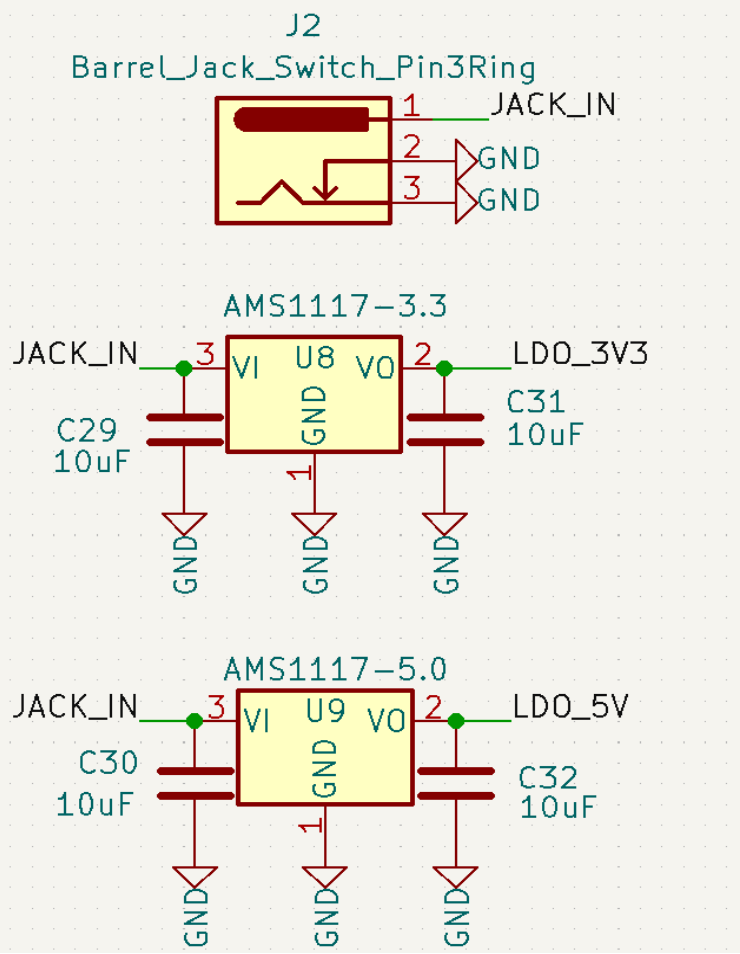
PORTD CONNECTIONS

- The figure below shows the various connections available for each of the PORTD pins.



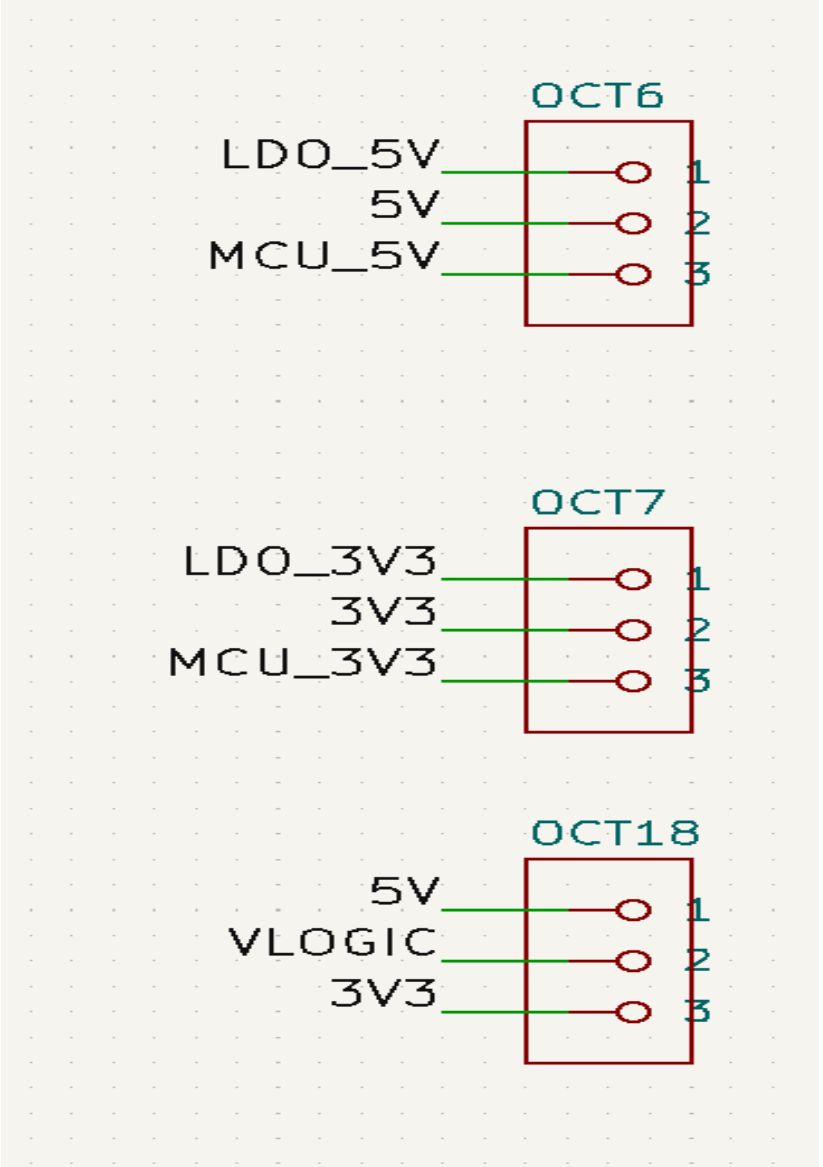
VOLTAGE REGULATORS

- The ntuAboard_G1 has 2 voltage regulators for producing DC voltages of 3.3 Volts and 5 Volts.
- The voltage regulators are powered by an external DC voltage applied to connector J2 and its value ranges from 7 to 13 volts.



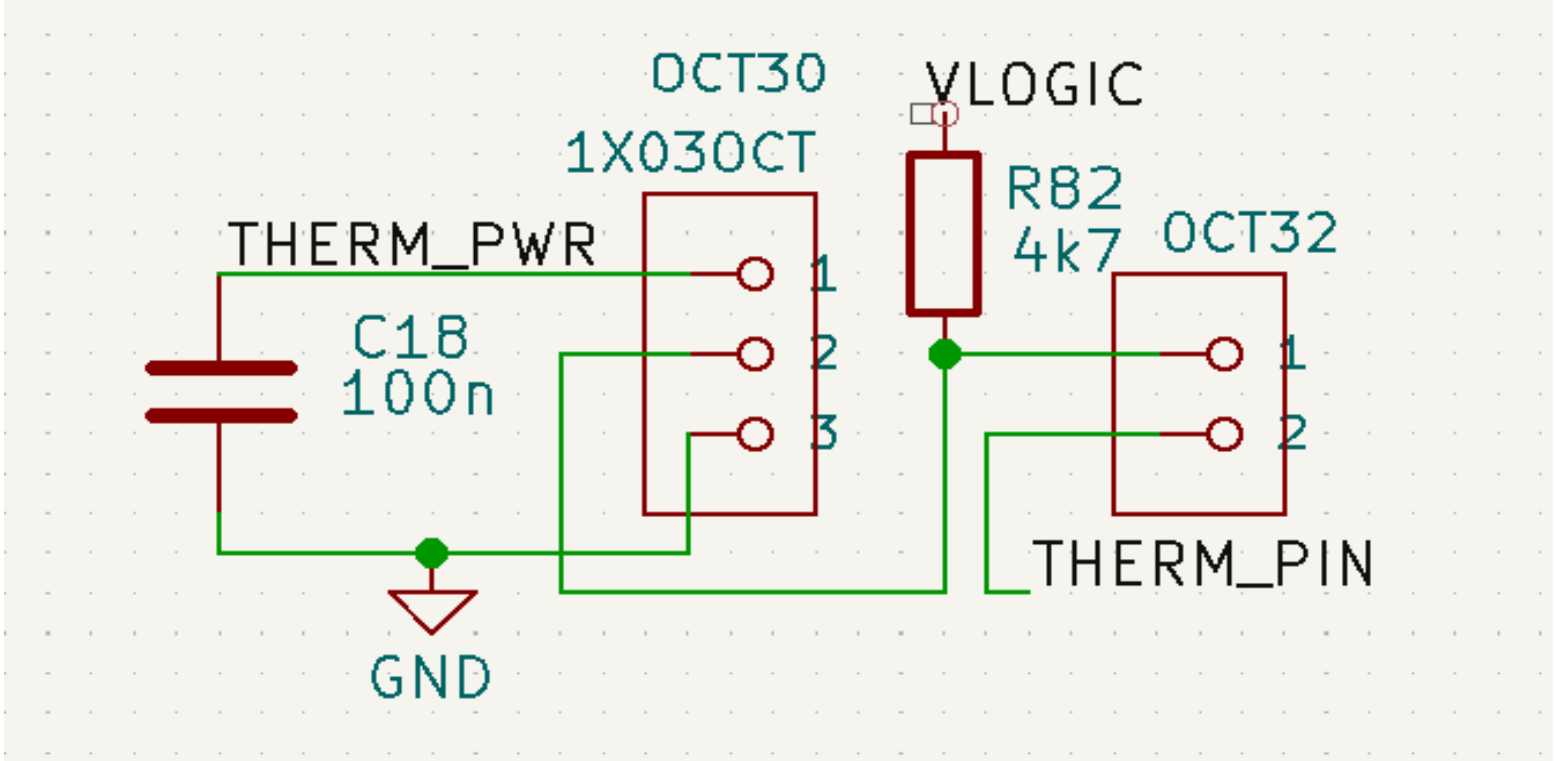
VOLTAGE SELECTORS

- The ntuAboard_G1 provides the option of selecting its supply voltage either from the ATmega328PB Xplained Mini card or from the 2 voltage regulators it integrates.
- This selection is made from the OCT6 and OCT7 connectors using appropriate jumpers, as shown in the adjacent figure.
- With connector OCT18 you can select whether the VLOGIC voltage of ntuAboard_G1 is 3.3Volt or 5Volt.



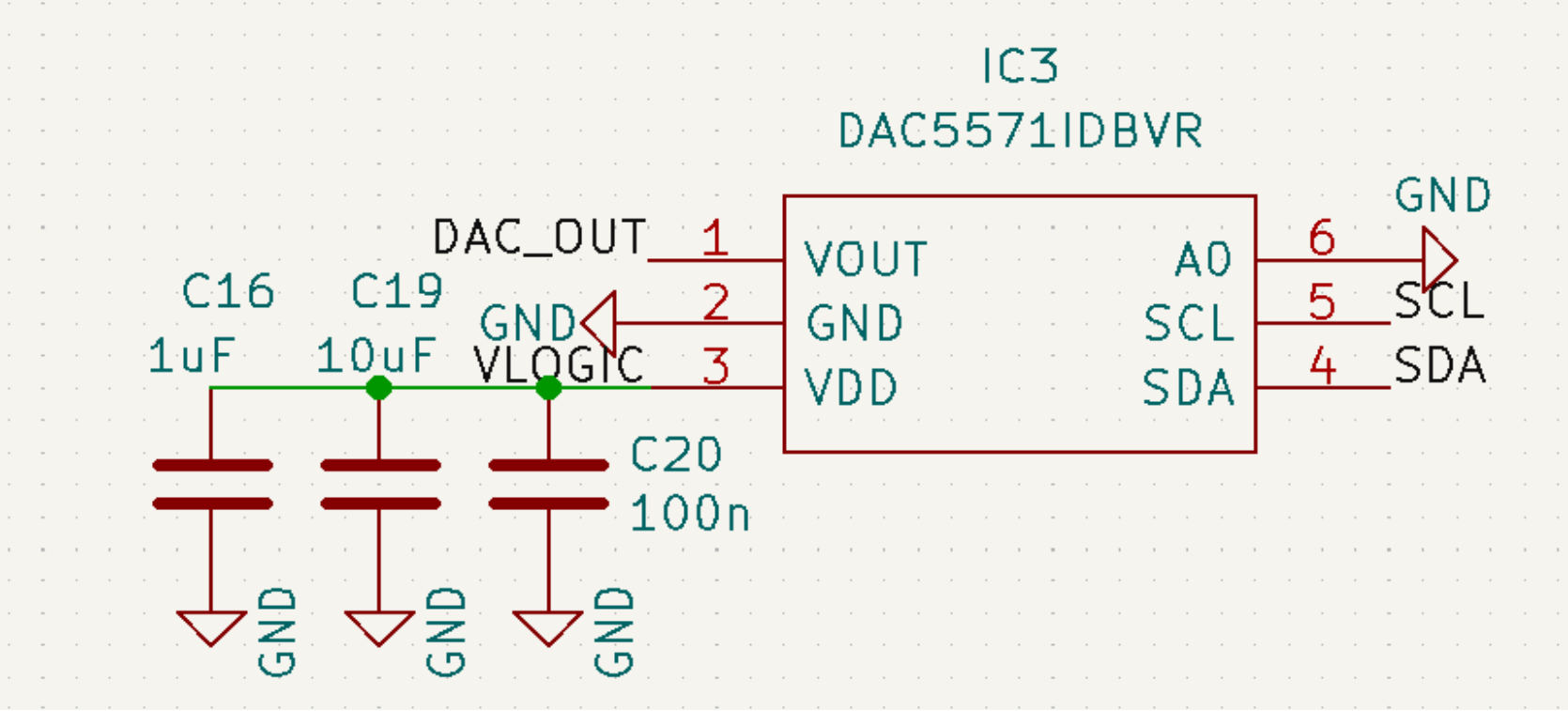
TEMPERATURE SENSOR

- A temperature sensor can be connected to connector OCT32. With connector OCT30 you can select whether this sensor is connected or not.



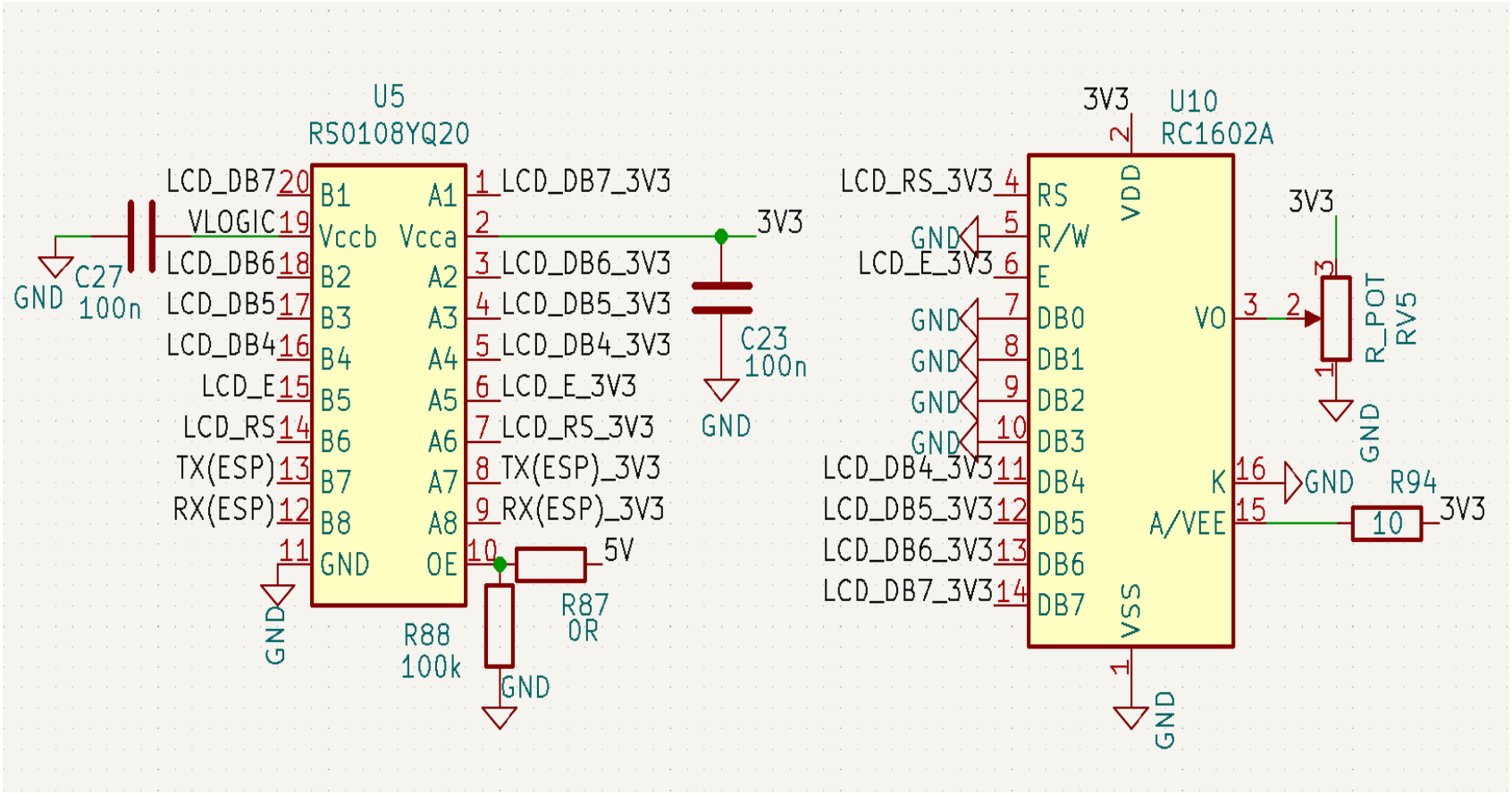
DIGITAL TO ANALONG CONVERTER (DAC)

- The ntuAboard_G1 has a digital to analog converter (DAC).
- The DAC converter is connected to the I2C port, using only two pins of the microcontroller, as shown in the figure below:



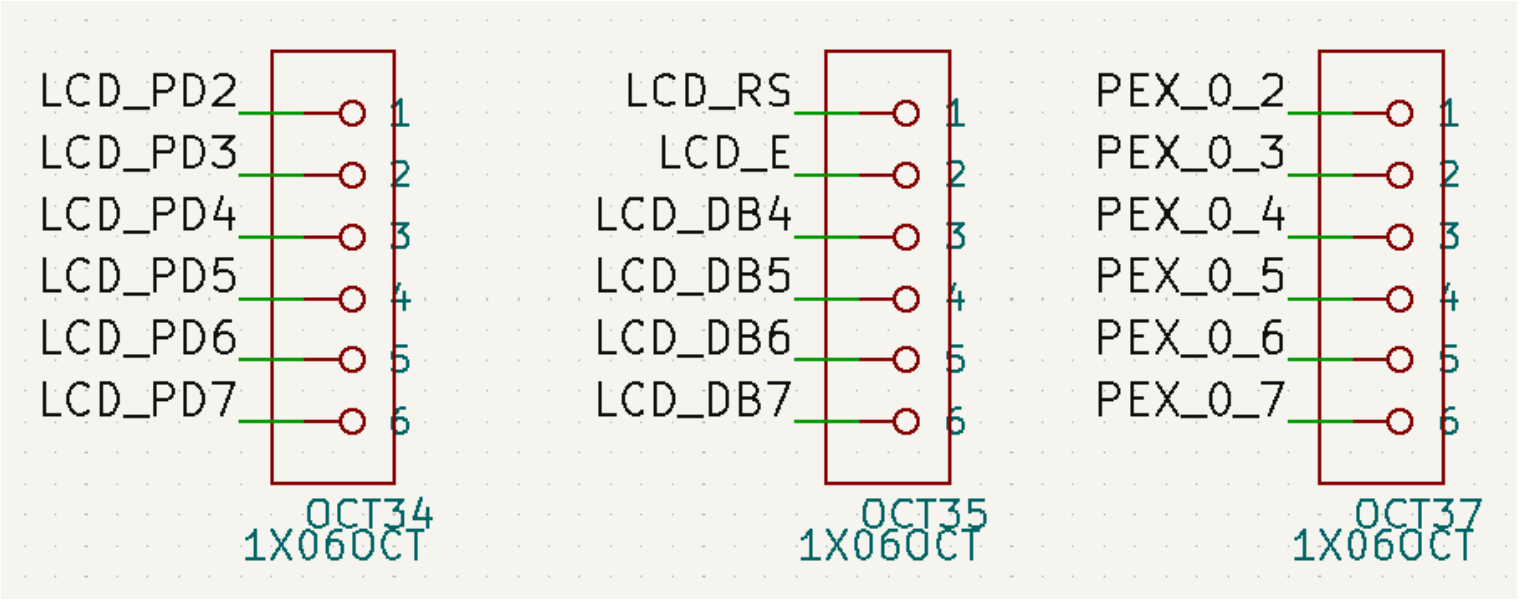
2x16 LCD

- The ntuAboard_G1 has a 2x16 character LCD screen.
- A voltage adjustment integrated circuit is inserted between the display and the microcontroller.



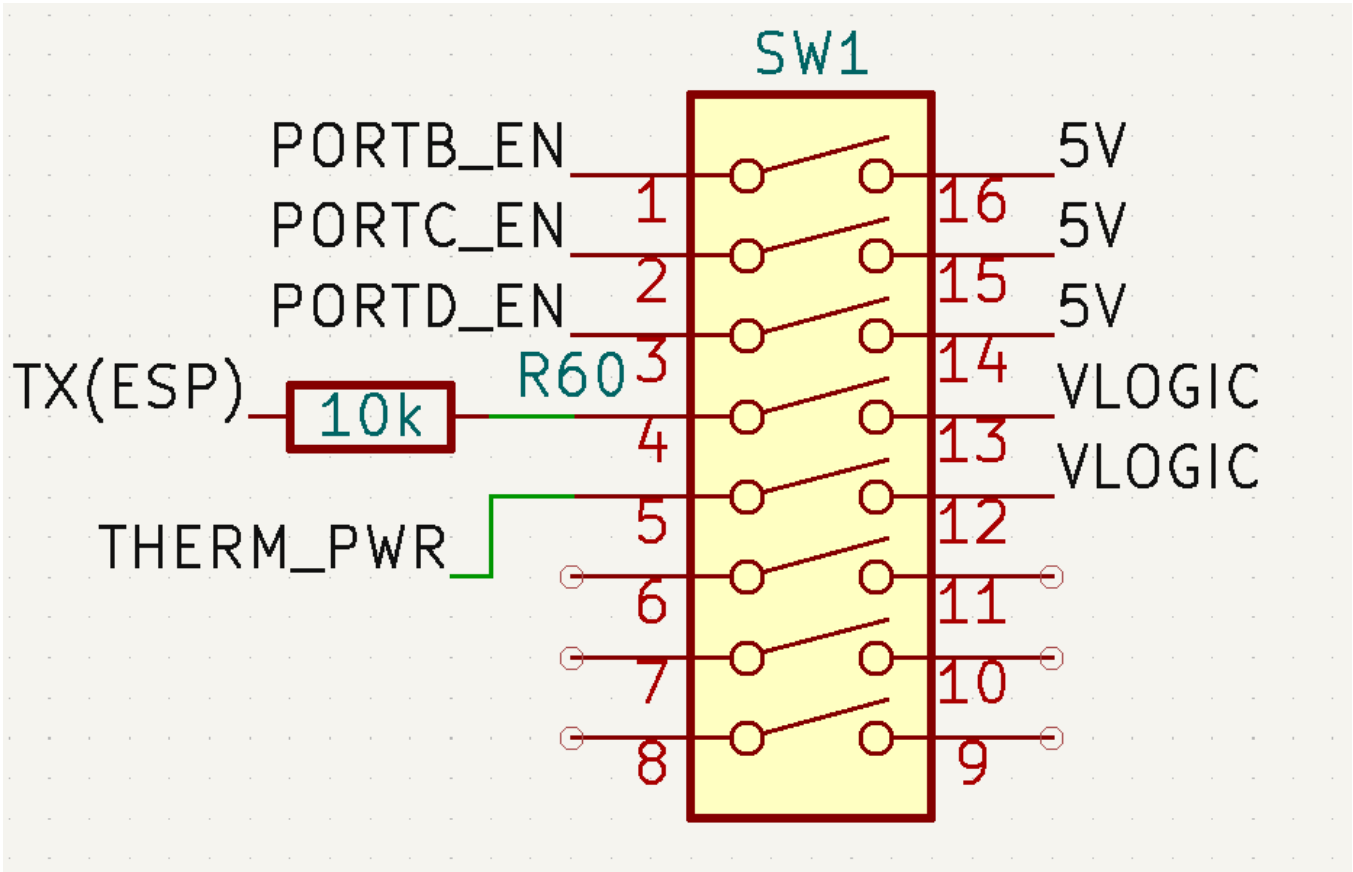
2x16 LCD

- By using the connectors OCT34, OCT35 and OCT37 we can choose whether the lcd screen will be connected to PORTD or PORT EXPANDER.

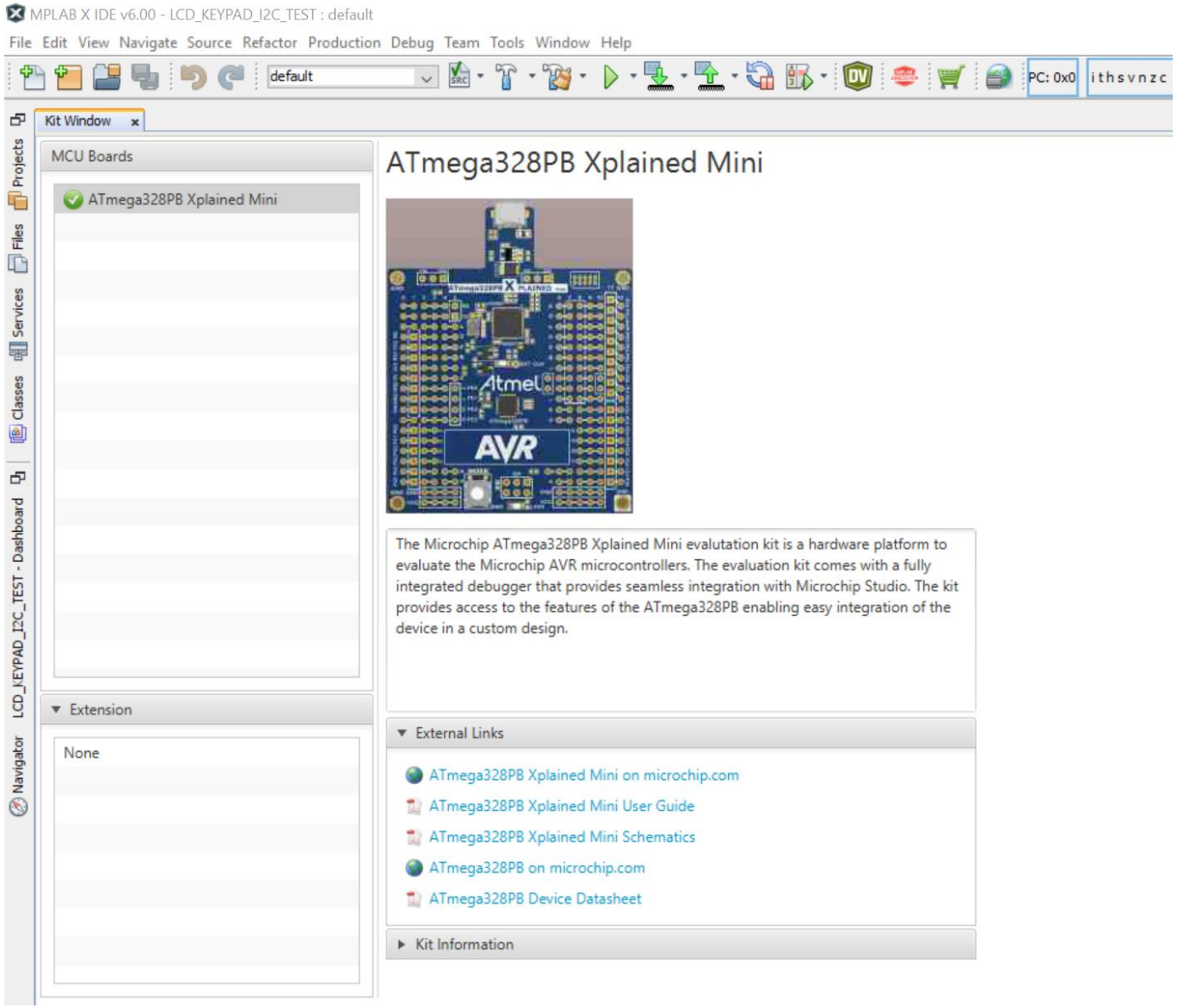


DIP SWITCHES

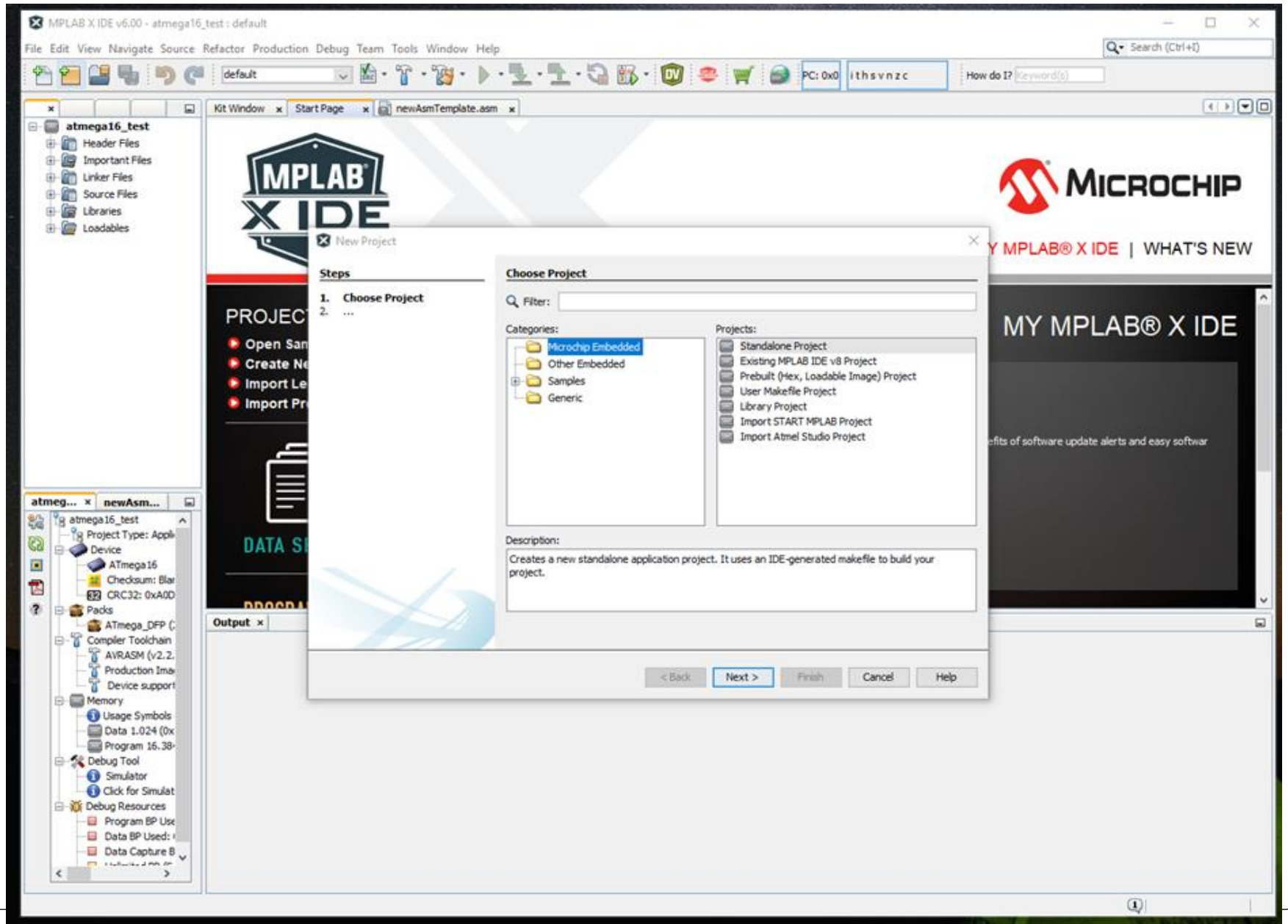
- With dip switches SW1 we can choose the connection or not of the leds of the three ports, of the TX(ESP) terminal and of the power supply of the temperature sensor.



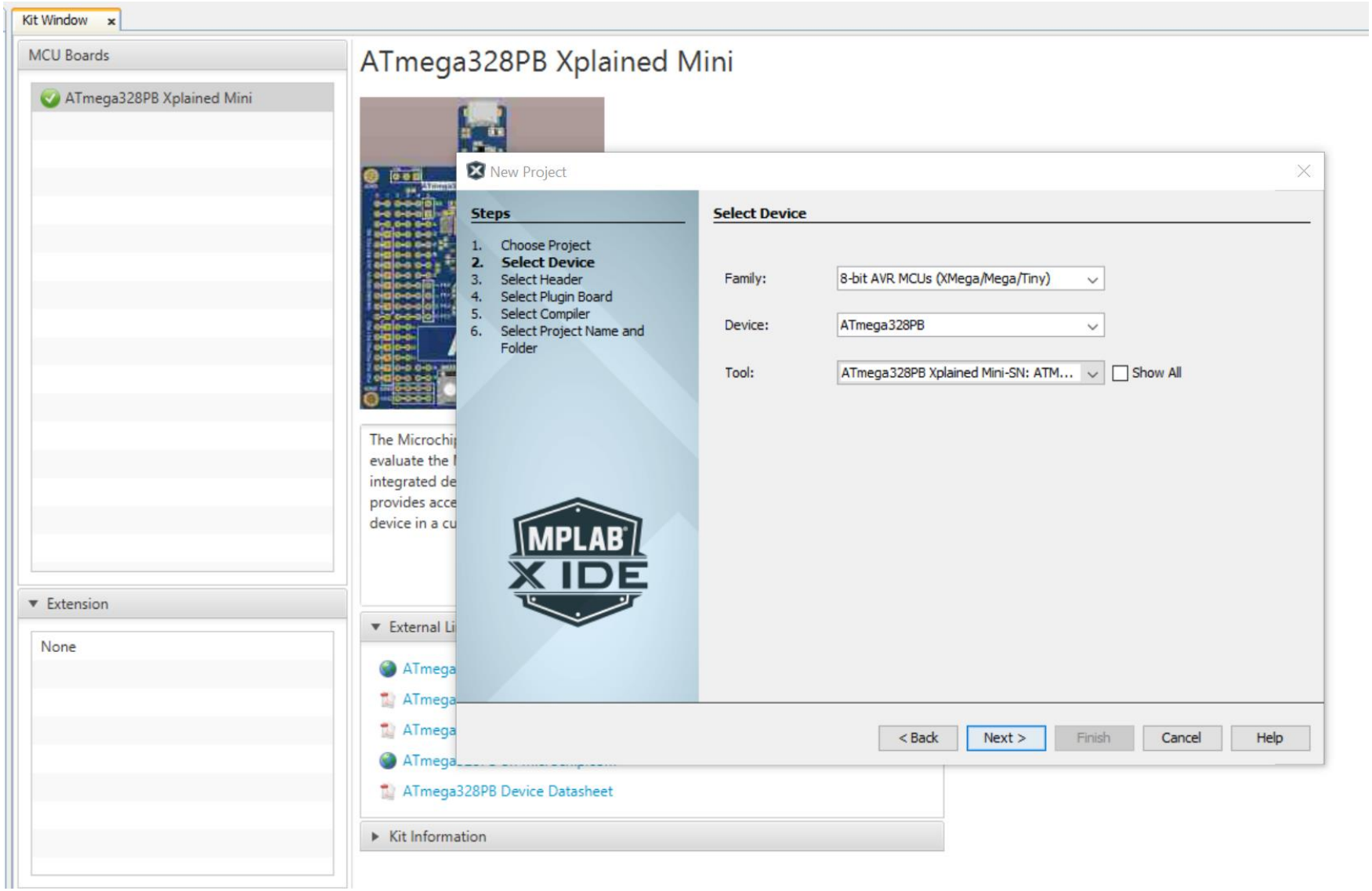
MPLAB X Kit Window



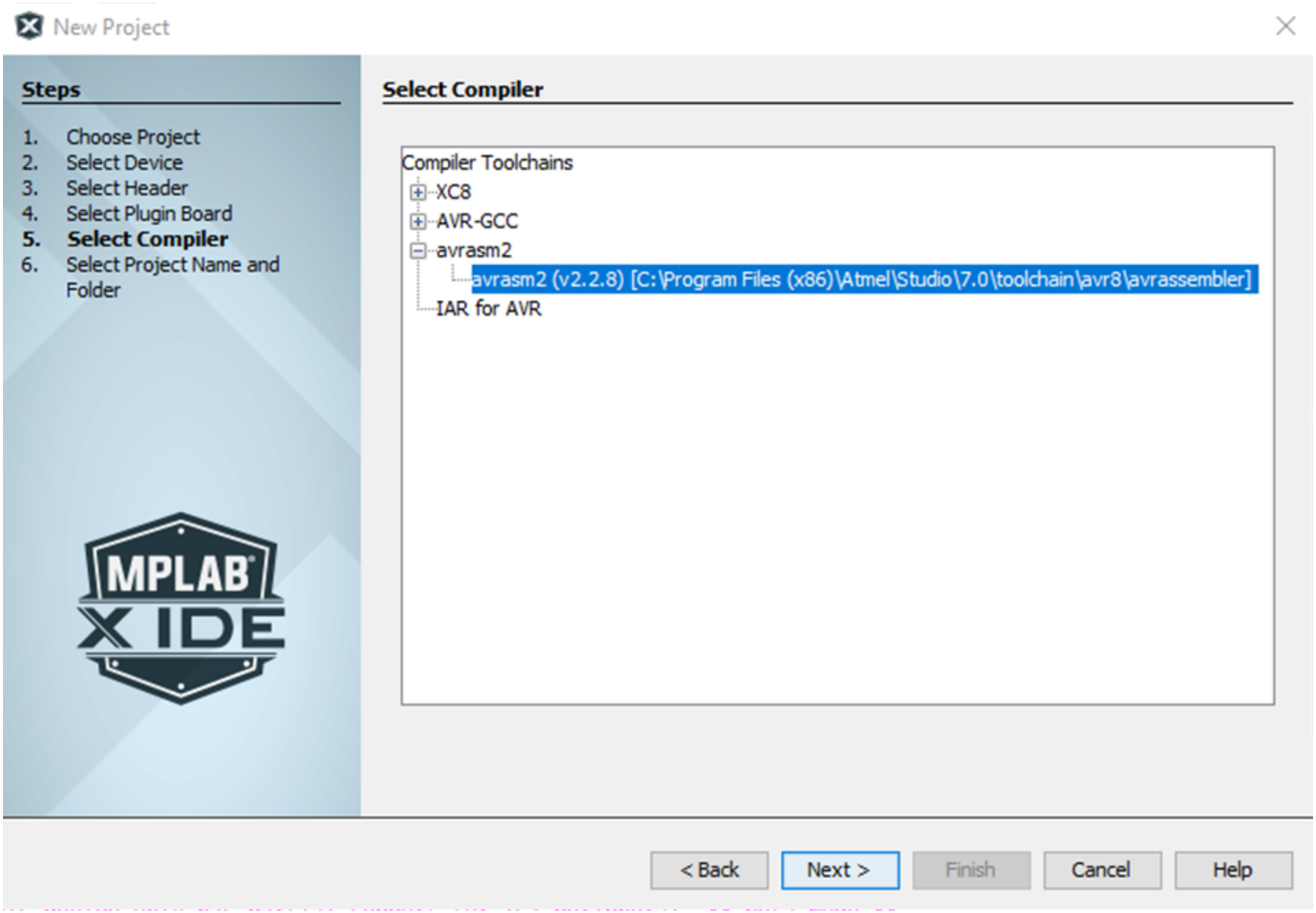
MPLAB X: File>New Project



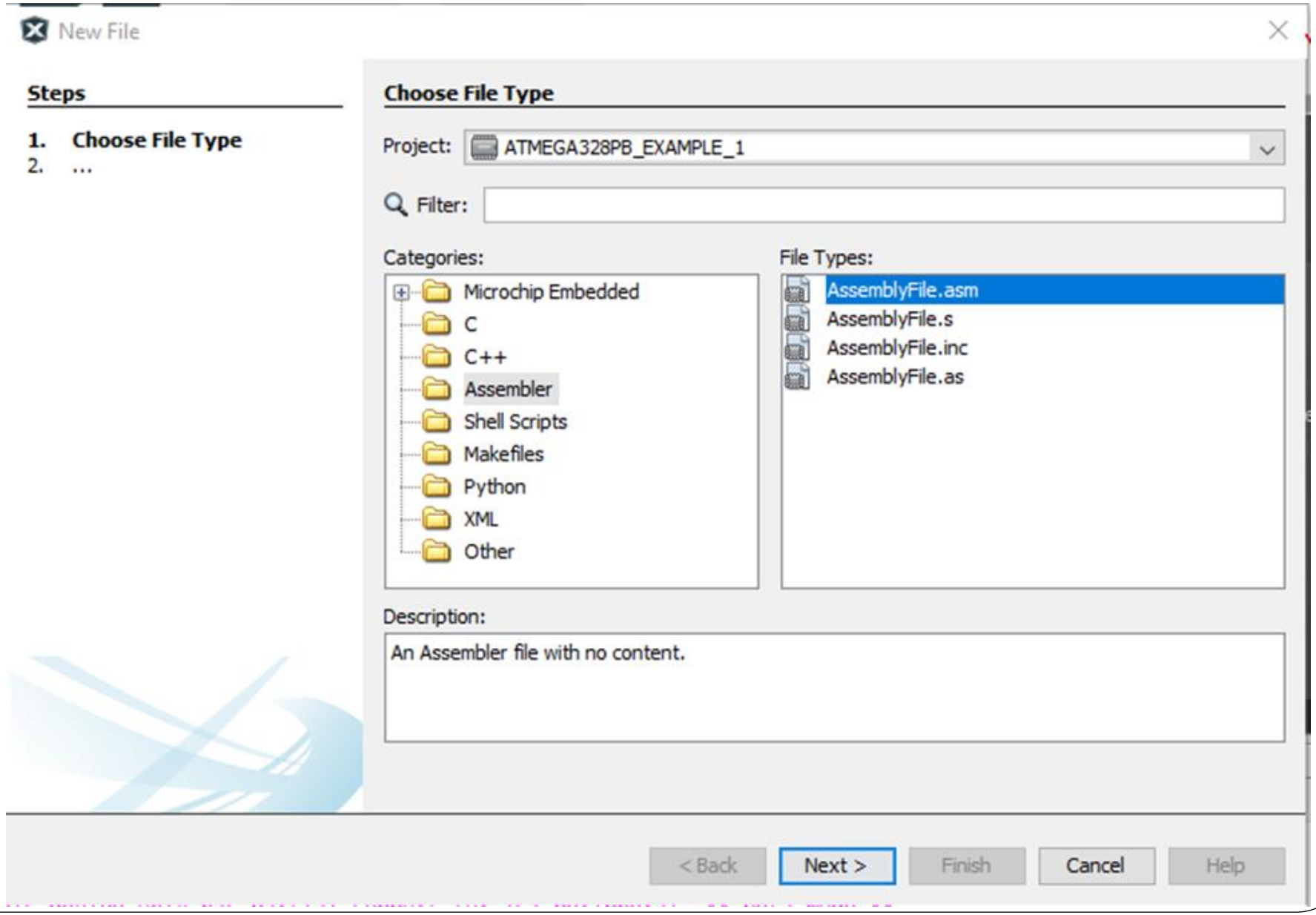
Select Device and Tool



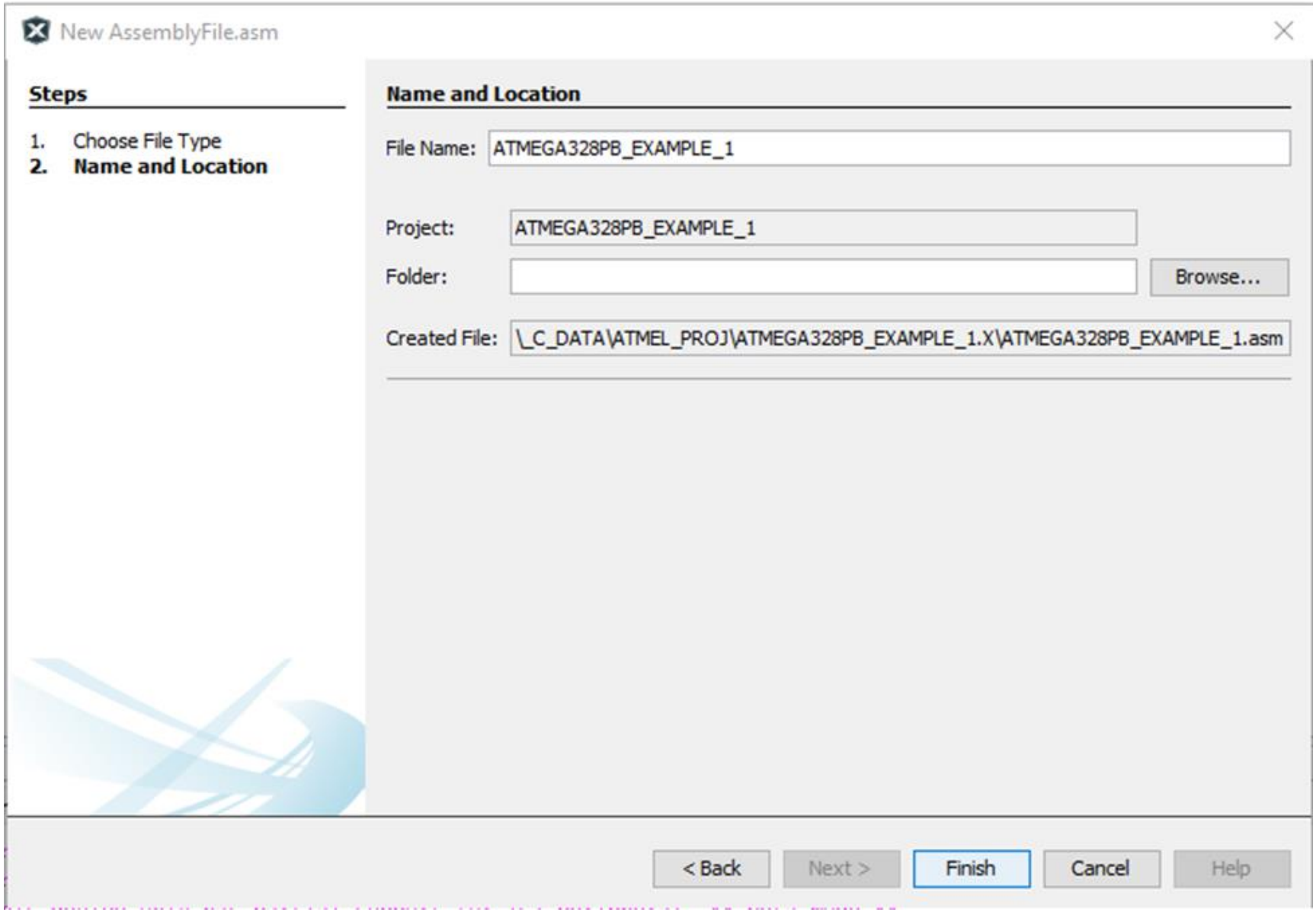
MPLAB X: Select assembler



MPLAB X: New File



MPLAB X: File Name and Location



Example of assembly code(page 1)

```
.include "m328PBdef.inc"    ;ATmega328P microcontroller definitions

; delay = (1000*F1+14) cycles (abougt DEL_mS in mSeconds)
.equ FOSC_MHZ=16          ;MHz
.equ DEL_mS=1    ;mS
.equ F1=FOSC_MHZ*DEL_mS

;Init Stack Pointer
    ldi r24, LOW(RAMEND)
    out SPL, r24
    ldi r24, HIGH(RAMEND)
    out SPH, r24

;Init PORTD as output
    ser r26
    out DDRD , r26

    ldi r24, low(F1)          ;
    ldi r25, high(F1)         ;Set delay

loop1:
    ser r26
    out PORTD, r26
    rcall delay_outer         ;  mS
    clr r26
    out PORTD, r26
    rcall delay_outer         ;  mS
    rjmp loop1
```

Example of assembly code(page 2)

```
;this routine is used to produce a delay 993 cycles
delay_inner:
    ldi r23, 247            ; (1 cycle)
loop3:
    dec r23                 ; 1 cycle
    nop                     ; 1 cycle
    brne loop3              ; 1 or 2 cycles
    nop                     ; 1 cycle
    ret                     ; 4 cycles


;this routine is used to produce a delay of (1000*F1+14) cycles
delay_outer:
    push r24                ; (2 cycles)
    push r25                ; (2 cycles) Save r24:r25

loop4:
    rcall delay_inner       ; (3+993)=996 cycles
    sbiw r24 ,1             ; 2 cycles
    brne loop4              ; 1 or 2 cycles

    pop r25                 ; (2 cycles)
    pop r24                 ; (2 cycles) Restore r24:r25
    ret                     ;4 cycles
```

Example in C language

```
#define F_CPU 16000000UL
#include <util/delay.h>
#include <avr/io.h>

#define DEL 1000U

int main(void)
{
    DDRD = 0xFF;
    while (1)
    {
    PORTD = 0x00;
    _delay_ms(DEL);
    PORTD = 0xFF;
    _delay_ms(DEL);
    }
}
```