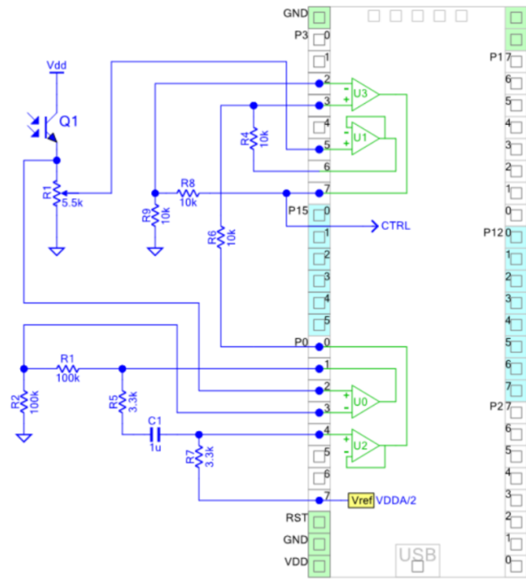
Annotation Library

1.0

Features



- Visualizes external connections to development kits
- Complements stock annotation components
- Small component size (~60% of standard size)
- Resolves names collision
- Doesn't consume system resources
- Doesn't affect run-time performance

General description

The Annotation Library(*) is a set of external components intended for documenting PSoC schematics. It consists of more than 40 components (Resistor, Capacitor, Opamp, etc.) designed for visualizing external connections to the PSoC prototyping kits and creating schematics. The Library complements Creator stock off-chip annotation components. It doesn't consume system resources, nor affects run-time performance of the project. The Library is not specific to PSoC4, 5 or 6 and can be used with any PSoC prototyping kit.

When to use Annotation Library

The Library was developed for documenting simple projects, which were build using PSoC prototyping kits and a breadboard carrying external components, – a configuration typically found in educational environment. The Library is particularly useful when compact placement is a premium; its components are typically smaller than the stock counterparts and were specifically designed to be used with annotation stubs for PSoC prototyping kits: CY8CKIT-042 Pioneer Board, CY8CKIT-044 and CY8CKIT-059 [1-4].

---

* Hereafter called the Library

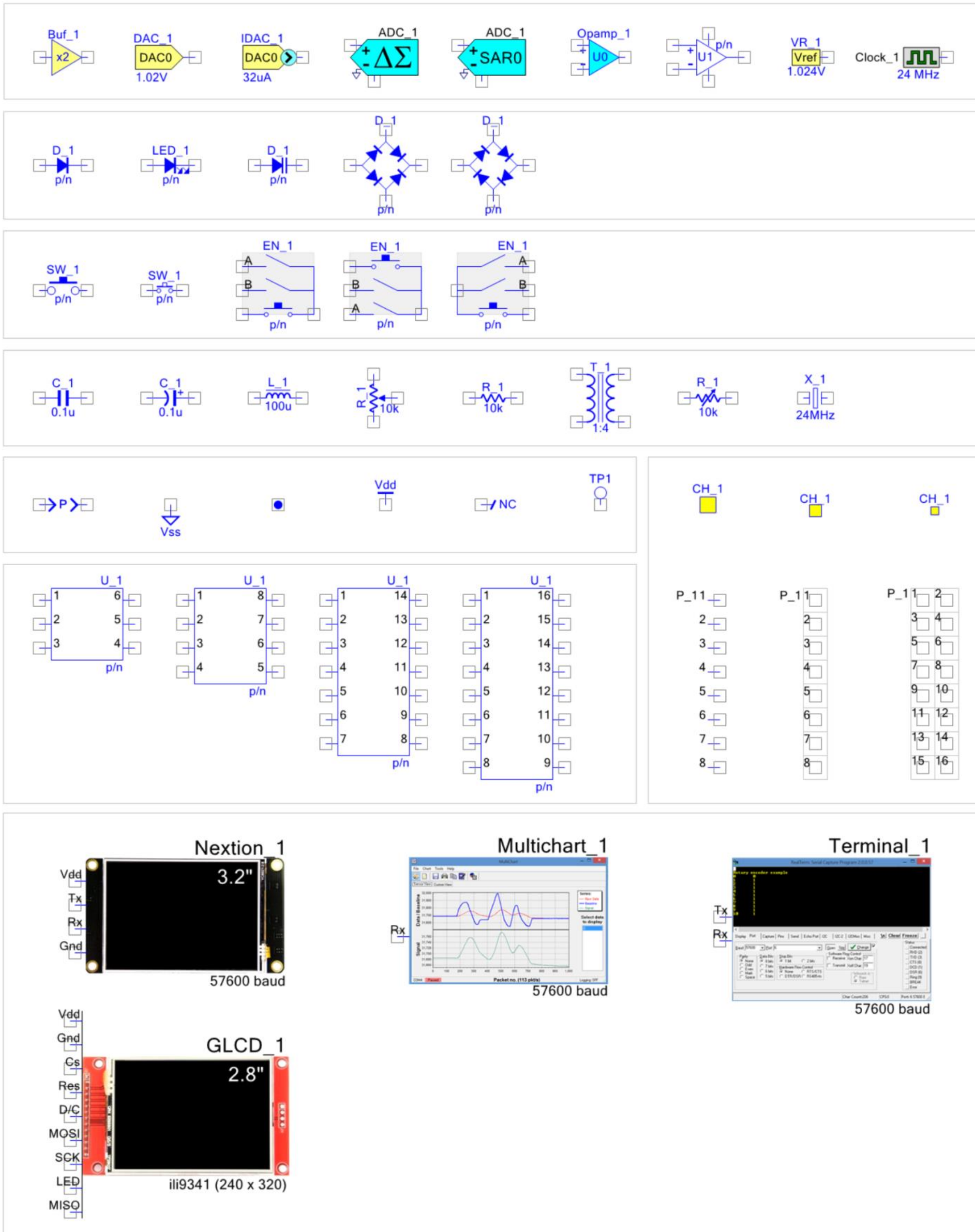


Figure 1. List of components included in the Library: Active, Diodes, Electro-mechanical, Passive, Power, Small Outline Package, Primitives, Interface.

Functional Description

Cypress Creator IDE offers various stock annotations components, which help visualizing project functionality. The Library complements stock components and addresses several issues:

- The Library components are smaller (typically 60% of the stock counterparts), which allows for denser placement on the schematic.

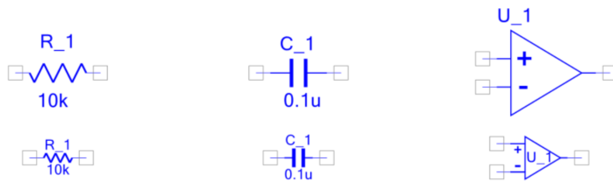


Figure 2. Component size comparison: top – stock components, bottom – Library counterparts.

- The Library resolves names collision, when same off-chip component has to appear in several instances in the project, for example on the schematic page and the annotation diagram page. By default, the component Instance Name can't be duplicated, automatically incrementing the name index (Figure 3a, b). Using Library resolves this problem by displaying user-assigned Net Name instead of the Instance Name (Figure 3c). See **Implementation** section for details.

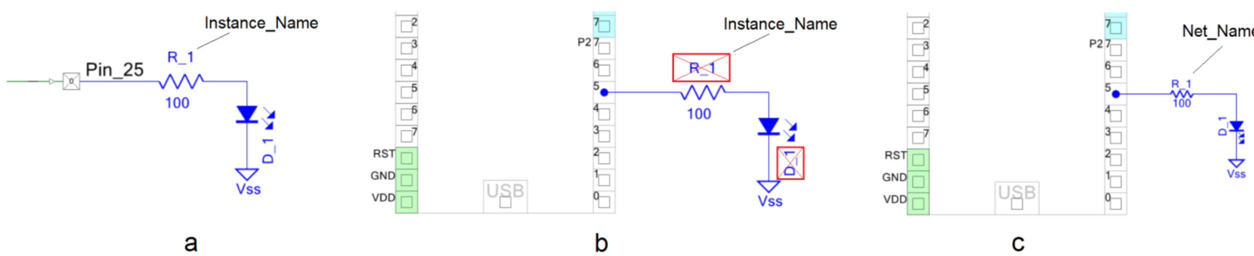


Figure 3. Resolving names collision using Library components: (a) project schematic, (b) annotation using stock components (names collision), (c) annotation using the Library components (names collision resolved).

- The Library adds several new components (Button Switch, Rotary Encoder, DAC, etc.), and primitives (wire break, join dot, no connect, color marker, etc.).

- The Library introduces extra component properties for record keeping (part number, supplier catalog number).

Implementation

Instance Name / Net Name

Unlike standard components, the Library parts can display either an Instance Name or a Net Name in the top identification label. The Instance Name is the unique identifier of the component, automatically assigned by the Creator IDE; only a single instance of the component with that Instance Name can exist in the entire project. Certain rules apply to the instances naming convention. To the contrary, the Net Name is user-selectable, and can be any string of characters. The Net Name does not have to be unique – it is user responsibility to keep track of components labels. For example, a Resistor displaying a Net Name “R1” can appear on several pages of the project without causing names collision. The Net Name may coincide with the Instance Name of the component, for example a Resistor with Instance Name “R_1” can also be given a Net Name “R_1”, etc.

Value / Part Number

The Library components falls into two major categories: those which have associated Value parameter (100kO, 100pF, 100uH, etc.), and others which haven’t. Passive components (Resistor, Capacitor, Inductor, etc.) traditionally display their values on the schematic. Other components (Diode, LED, Opamp, etc.) have no Value parameter; instead, they customary display their Part_Number. See **Parameters and Settings** section for details.

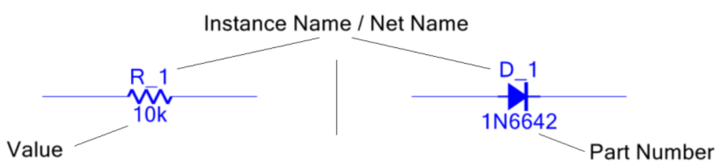


Figure 4. Appearance difference between components: Resistor (value visible), Diode (part number visible).

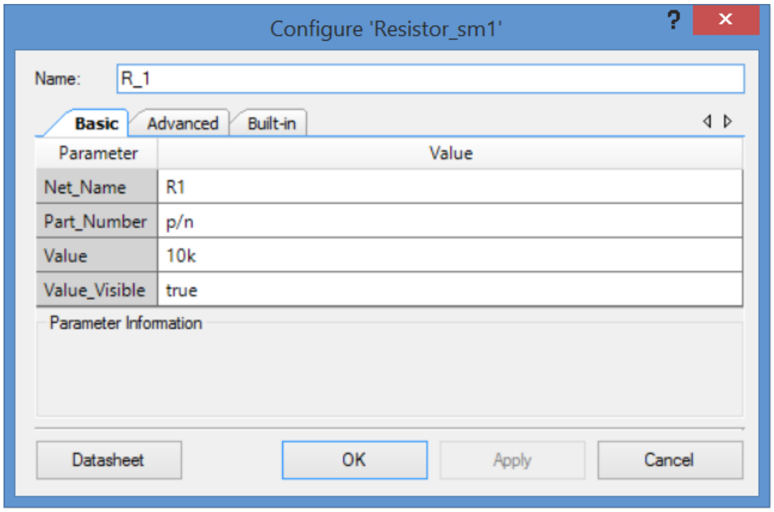
Component naming convention

In Creator, all component names in the Component Catalog must be unique. To distinguish from the existing stock components with same names (Resistor, Capacitor, etc.) all Library components are identified by appending a suffix “_sm1”, uniquely indicating its affiliation with the Library(*).

* small type 1

Parameters and Settings

Typical Basic dialog provides following parameters(*):



Net_Name (string)

Component Net Name. Displays user-defined name of the component (for example "R1"). It is not auto-incremented and may differ from the Instance Name. Visibility of this parameter is controlled by Display_Name option in the **Advanced** dialog. The Net Name can be displayed on component instead of the Instance Name (default). The Net names are not automatically checked for duplicate instances, it is user responsibility to keep track of them.

Part_Number (string)

Manufacture part number (for example "LM741"). The visibility of this parameter is controlled by Part_Number_Visible option. Passive library components (Resistor, Capacitor, Inductor, etc.) have this parameter hidden, only showing it in the popup window on mouse hover.

Part_Number_Visible (bool)

Sets visibility of the Part_Number. Default value is True. This setting is not available for passive components (Resistor, Capacitor, Inductor, etc.), where Part_Number is always hidden.

* The Dialogs shown are for the Resistor component; actual parameter list varies between components.

Value (string)

Component value (for example “10k”). This setting is specific for passive library components (Resistor, Capacitor, Inductor, etc.), which display their Value instead of the Part_Number. Visibility of this setting is controlled by parameter Value_Visible.

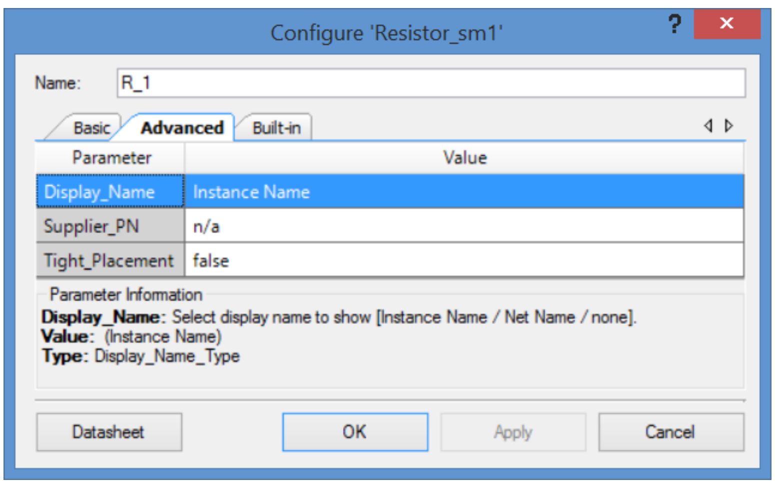
Value_Visible (bool)

Sets visibility of the component value. Default value is True.

Color [none/white/yellow/red/green/blue/lime/fuchsia/cyan]

Sets color appearance for certain components (TestPoint, ColorMarker, Terminal). It is useful for highlighting a component on the schematic, indicating LED color, scope probe color, etc. The “none” option provides transparent background.

Typical Advanced dialog provides following parameters:



Display_Name [Instance Name / Net Name / none]

Selects which name will be displayed by the top label on the component. Default is Instance Name (as above, “R_1”). Alternatively, a component can display a Net Name, which is user-assigned and can be any string of characters (for example, “R1”). If none option is selected the label is hidden. See also **Functional Description** section for details.

Supplier_PN (string)

Component supplier (distributor) catalog part number. For example, a Digi-Key catalog part number. This parameter provided for record-keeping. It is visible only in a popup window on the mouse hover.

Tight_Placement (bool)

Forces tight placement of the component labels (Instance Name, value, part number, etc.) to fit smallest footprint. This option is useful for tight parallel placement of the components. Default value is false. The effect of this parameter on component's visual appearance is provided in Table 2 (Appendix 1).

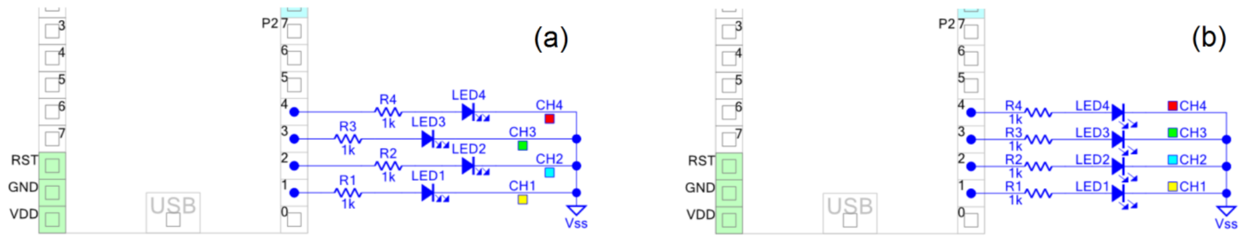


Figure 5. Effect of Tight_Placement option on component visual appearance: (a) default, (b) tight.

Orientation [vertical/horizontal]

Sets component orientation. Default value is vertical. This parameter is available only for few basic components (Ground, Power, TestPoint), and allows to rotate a shape without rotating its label. The difference with standard rotation of a shape is that it preserves component label orientation (Figure 6).

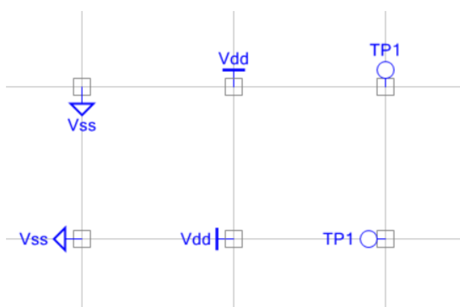


Figure 6. Effect of the Orientation option: top line – vertical, bottom line – horizontal.

Application Programming Interface

The component does not have associated API.

Resources

The Annotation library doesn’t consume any hardware resources. The Annotation Library is not device-specific and can be used in conjunction with any of PSoC4, PSoC5 or PSoC6 kits.

Performance

The Annotation Library does not affect PSoC run-time performance.

Application examples

Several application examples of the Annotation Library are provided in the **Appendix 2**.

Component Changes

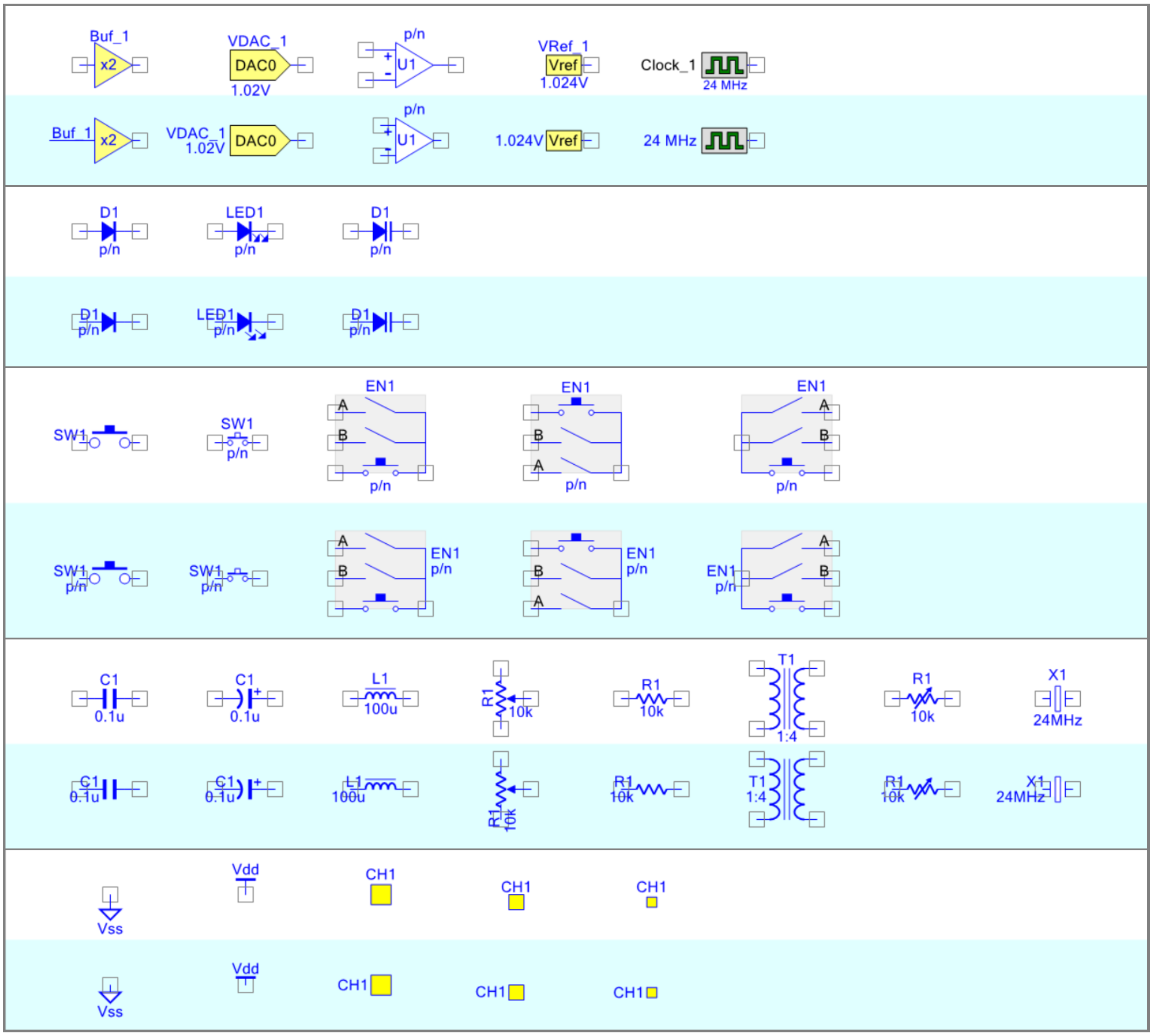
Version	Description of changes	Reason for changes/impact
0.0	Version 0.0 is the first beta release Library.	
1.0	The Library is reworked entirely. New properties structure is based on a single unified component stub. Components are assigned new names. New components added. Location path is changed.	

References

1. KIT-042 is part of the KEES library by Chris Keeser, https://github.com/salsasteveold/GlowTime/tree/master/KEES_Library.cylib
2. CY8CKIT-044 annotation stub, https://community.cypress.com/thread/44492
3. Annotation library v0.0, https://community.cypress.com/thread/12559
4. CY8CKIT-059 annotation stub, v1.0, https://community.cypress.com/thread/48012

Appendix 1

Table 1. Effect of the Tight_Placement parameter on component visual appearance. Top – default symbol appearance; bottom – with tight placement enabled.



Appendix 2

This section shows some typical application examples of the Library. They are illustrating a general idea and not directly related to the actual projects.

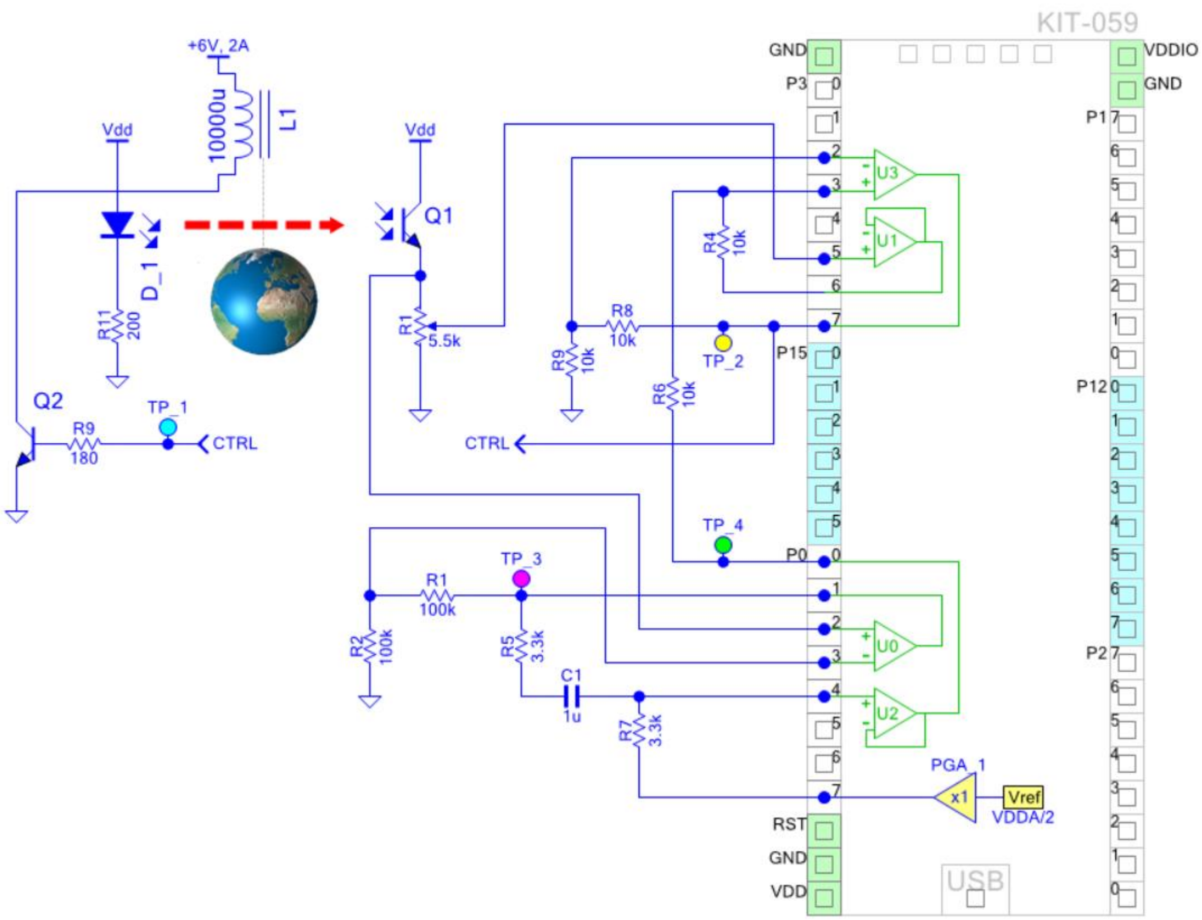


Figure 7. Annotation example of the all-analog PID-controlled levitation system(*) using KIT-059 stub, and a mix of the Library and stock annotation components. In this project, an optical sensor and a solenoid were used to levitate small plastic Globe with a magnetic core. The wiring diagram shows external connections to the CY8CKIT-059 prototyping board, while KIT-059 annotation stub shows board i/o pins to the PSoC5LP internal opamps. Test Points colors (TP_1-4) are set to match the scope display colors for easy identification of the signals. Note that in order to build this circuit, bypass capacitors C7, C9, C12 and C13 must be removed from the PCB board.

* All-analog PID control implemented using PSoC5LP prototyping kit CY8KIT-059. YouTube: https://www.youtube.com/watch?v=7FBw8-PBIb0

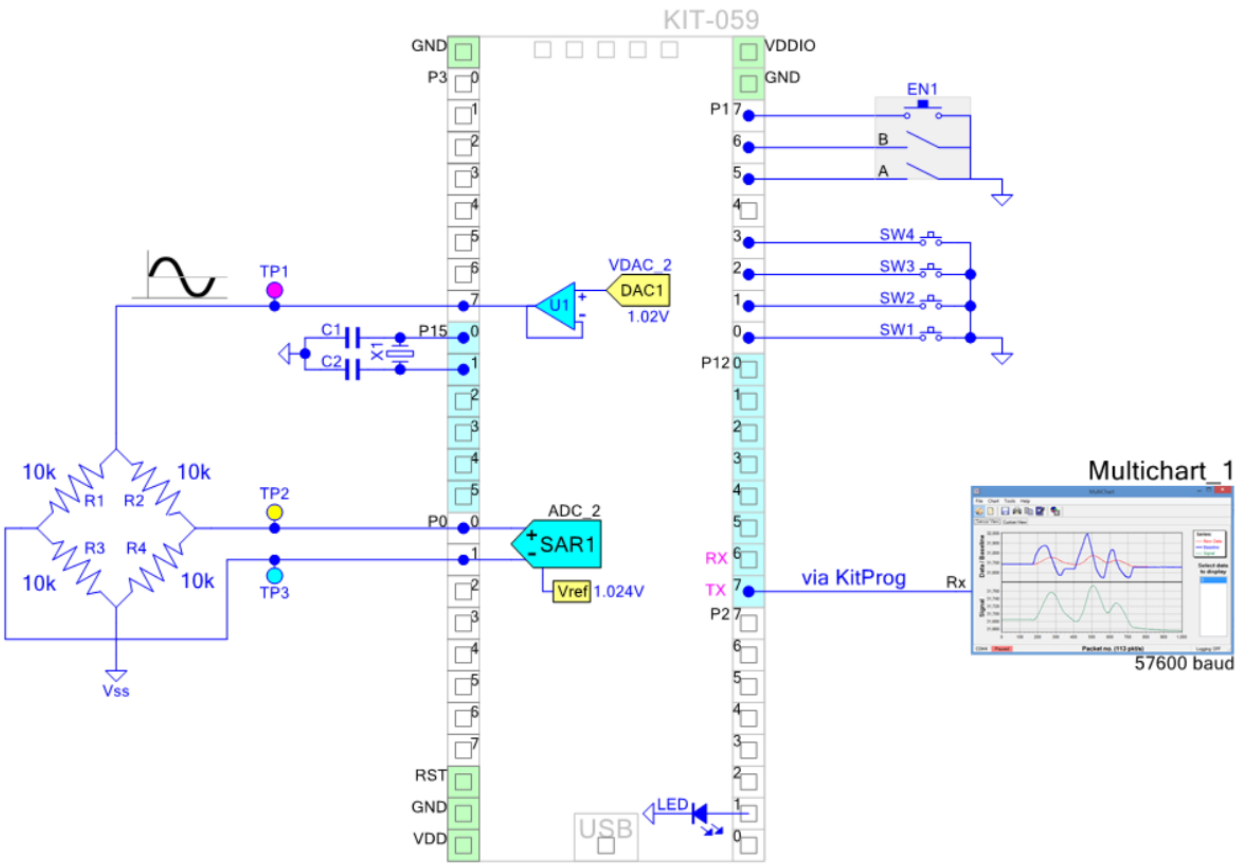


Figure 8. Annotation example of the resistive sensor system utilizing KIT-059 stub and mix of the stock and Library components. It depicts several off-chip components (resistive bridge sensor, Capacitor, XTAL), and on-board components (SAR_ADC, VRef, Opamp, VDAC, LED), which are used in the project for excitation of the sensor bridge and detection of the response signal. Other external parts (Rotary encoder and Button Switch) are used for updating system parameters. Visualization of the output data is performed using the MultiChart software running on the host computer, which communicates with PSoC UART Tx Pin_12[7] through the USB-UART interface built into the KitProg. The color of Test Points (TP1-3) is set to match the test scope display colors for easy identification of the signals.

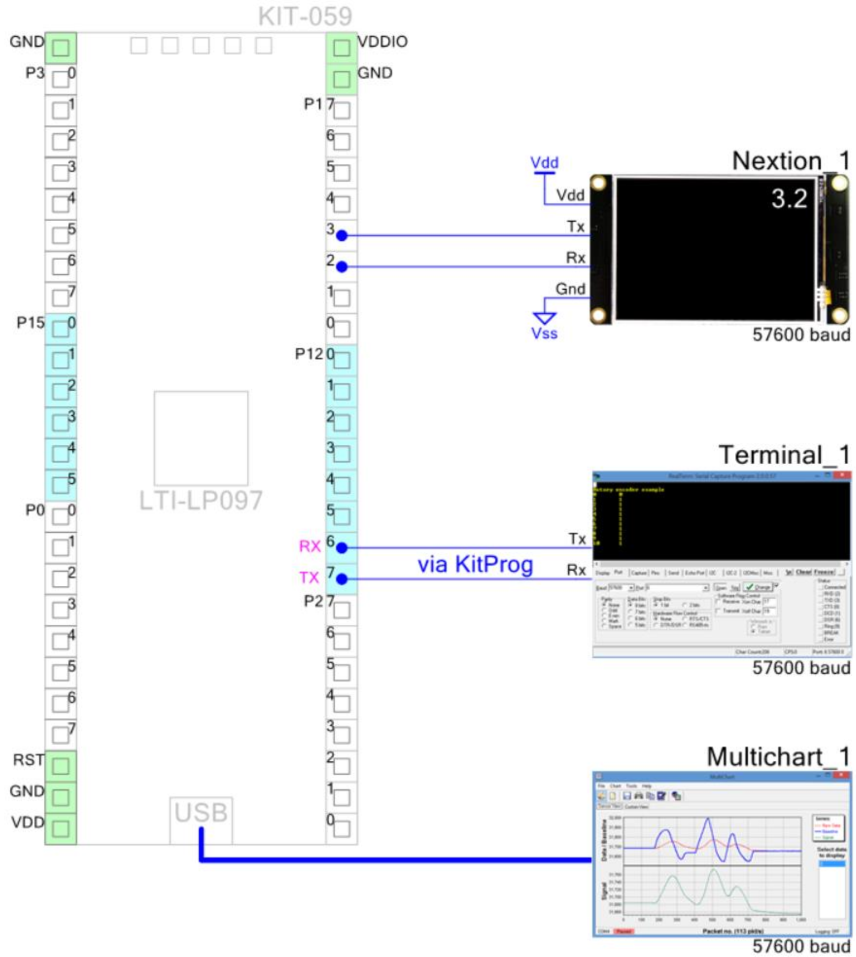


Figure 9. Annotation example of the bi-directional UART communication project, utilizing CY8CKIT-059 prototyping board and Nextion graphic display. In addition to the Nextion display, the project keeps bidirectional communication with the text terminal (RealTerm) on the host computer using USB-UART interface built into the KitProg. Chart plotting software (MultiChart) receives data stream from PSoC USB port.

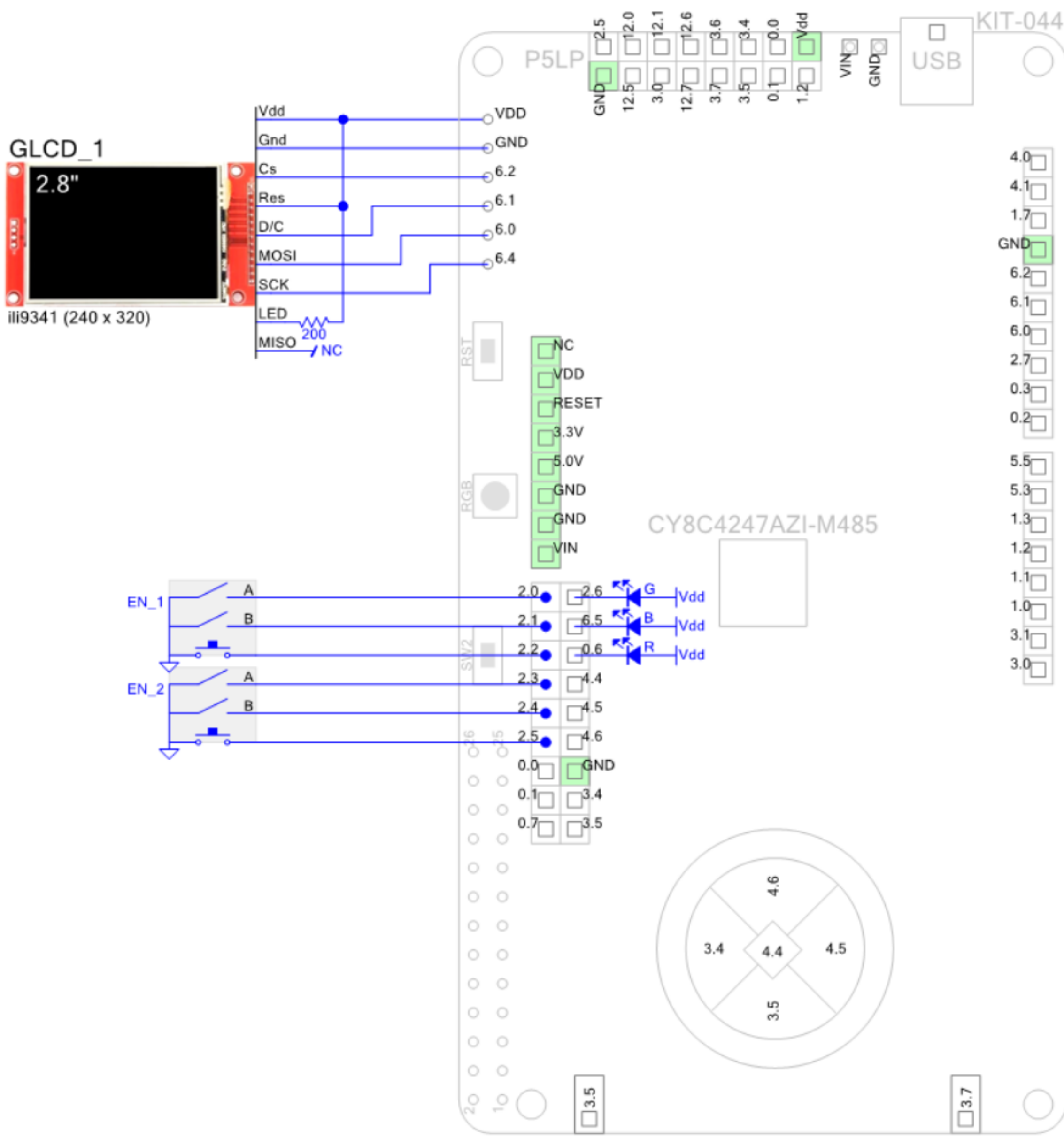


Figure 10. Annotation example of the GLCD ili9341 connection to the PMOD terminal of PSoC4 M-series Pioneer board (CY8CKIT-044). A pair of the rotary shaft encoders with push button switches is used for system parameters update. The annotation diagram utilizes some Library components (GLCD_9341, Rotary Encoder) and KIT-044 annotation stub [2] to display external connections to the prototyping kit.

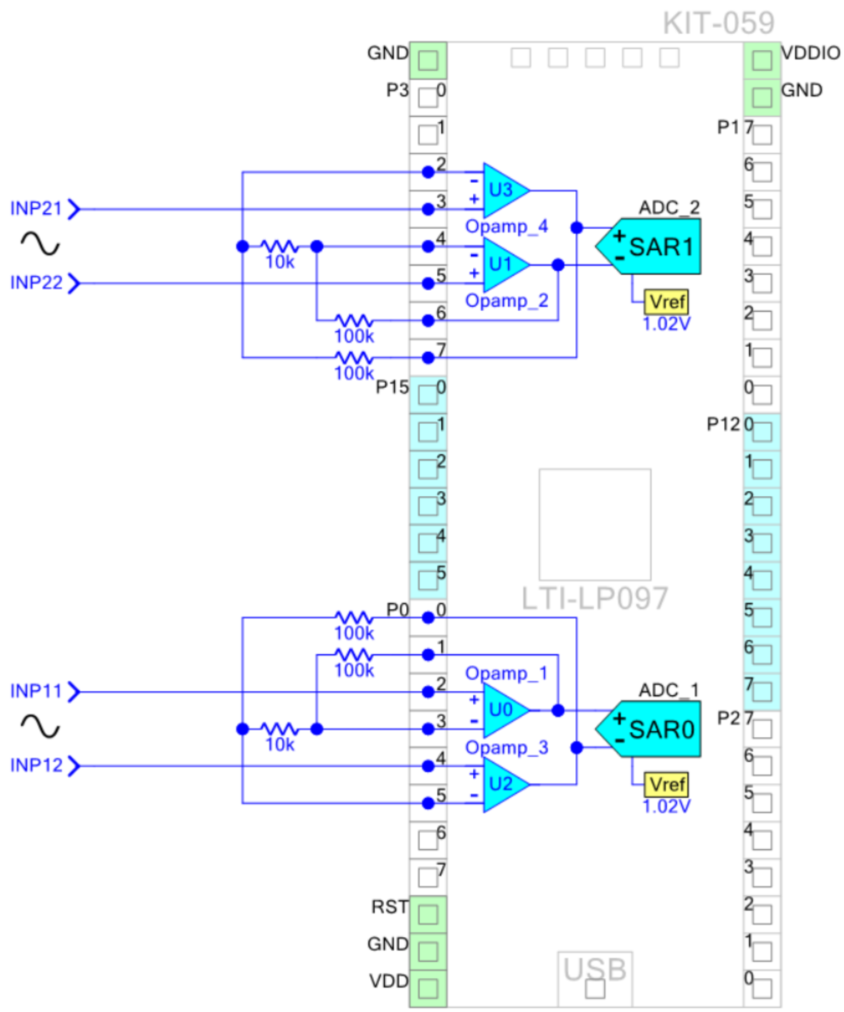


Figure 11. Example of the instrumentation amplifier implementation using CY8CKIT-059 prototyping kit. In addition to the KIT-059 stub, it utilizes some active components (OpAmp2, SAR_ADC, VRef) to visualize PSoC5 on-board hardware parts used in the project, as well as external passive components (Resistor, Potentiometer, Wire Break) to complete the circuit. Note that in order to build this circuit, bypass capacitors C7, C9, C12 and C13 must be removed from the PCB board.

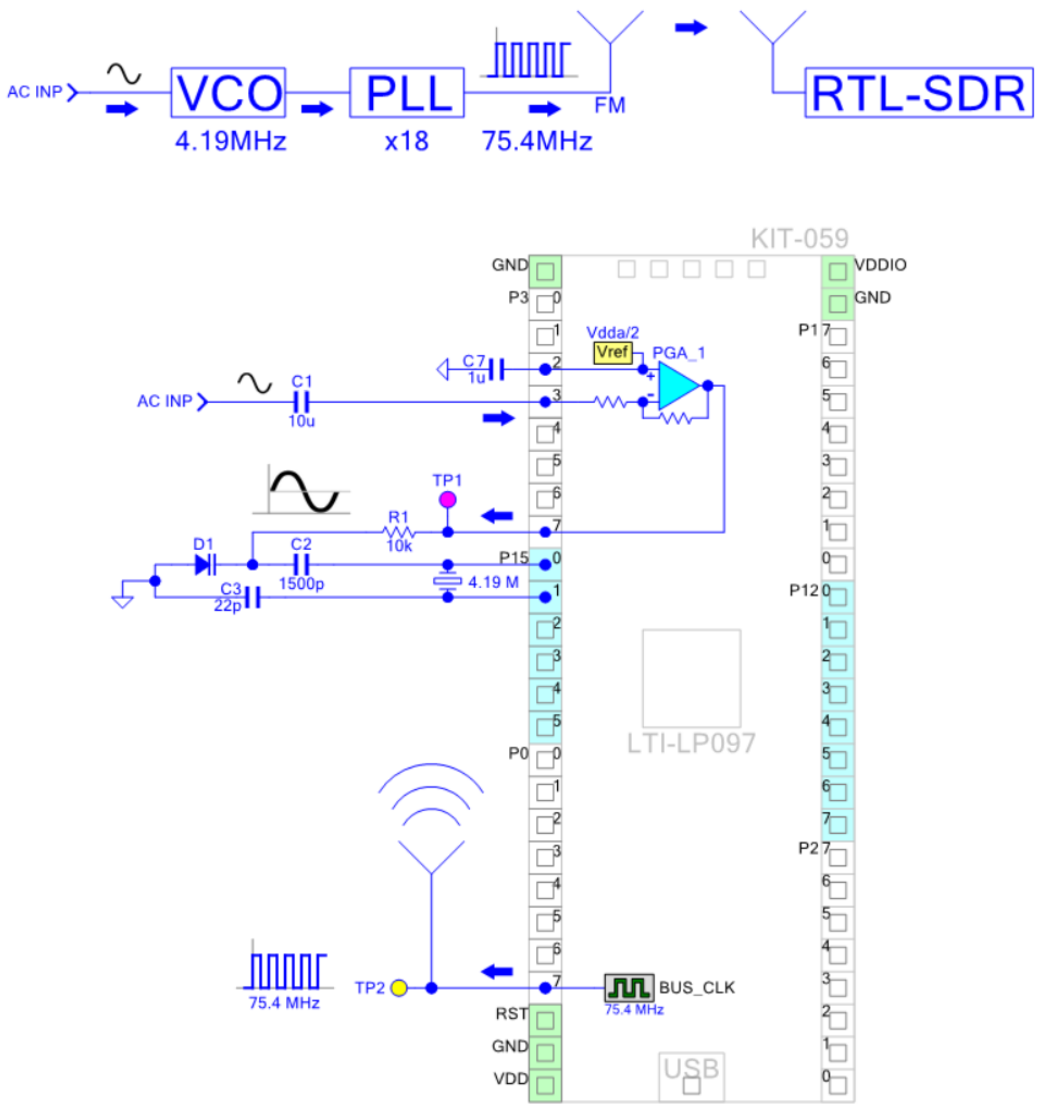


Figure 12. Annotation example of the FM radio transmitter[*] using CY8CKIT-059 prototyping kit. It uses direct clock connection to the digital output pin to produce FM radio signal on the BUS_CLK frequency. PSoC5 PLL is locked to the external 4.19 MHz ceramic resonator, connected between Pin_15[0] and Pin_15[1], which frequency is modulated using varactor diode D1, controlled by the external audio signal. PSoC5's PLL fast locking time (250 us) is sufficient to follow input signal at audio frequencies range. In addition to the KIT-059 stub, project annotation utilizes few active components (OpAmp2, Vref, Clock) to visualize PSoC5 on-board hardware, and some external components (Resistor, Capacitor, line Break, TestPoint, XTAL, Varactor) to complete the circuit. On-board bypass capacitors C7 is shown outside of the PCB board for clarity.

[*] Simple FM audio transmitter using PSoC5: https://community.cypress.com/thread/42364

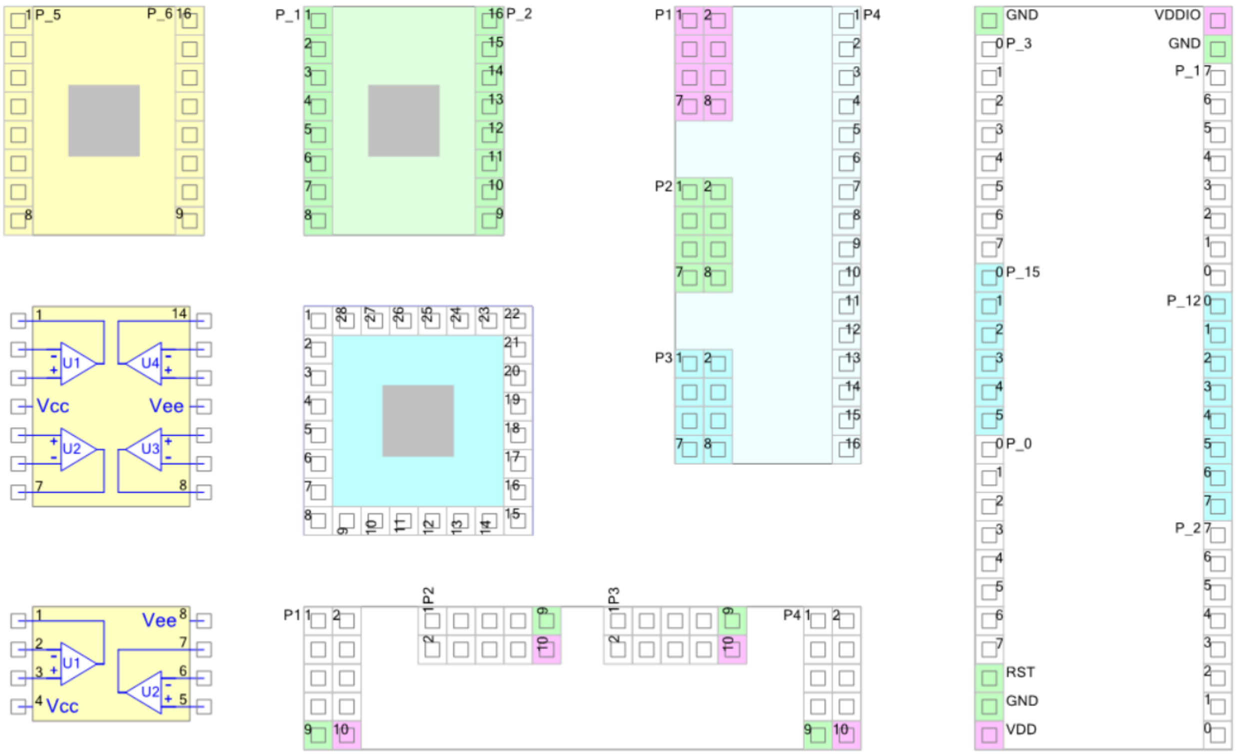


Figure 13. Various examples of usage of the Terminal and Connector primitives. The 1x8 and 2x8 Terminals can be used for annotating prototyping boards and custom PCB interface boards. Multiple Terminals can be combined to extend the number of pins up to 255. The Connector component may be used in similar ways, e.g. for creating a chip macro.

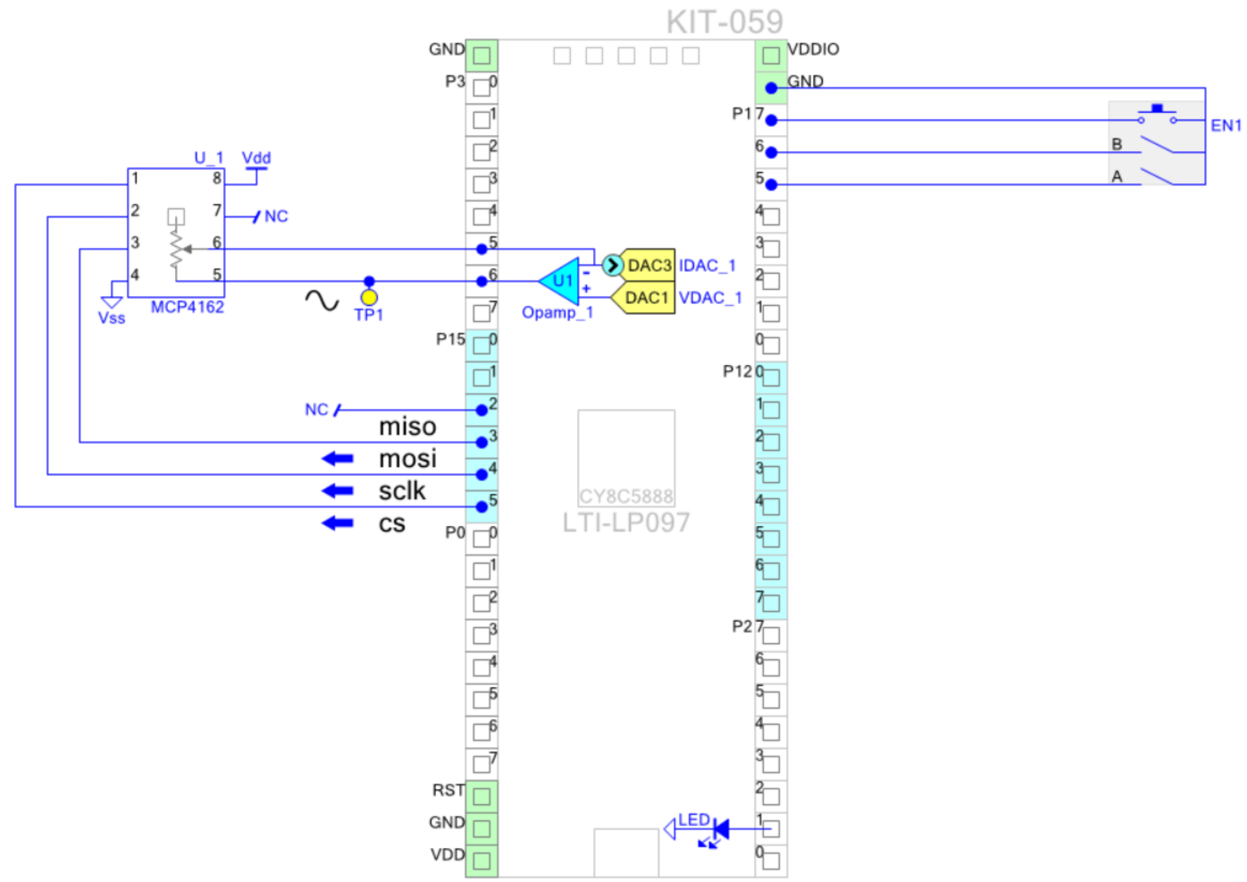


Figure 14. Annotation example of the output signal amplitude control using digital potentiometer (MCP4162). The Opamp_1, IDAC_1 and digital potentiometer U_1 form a transimpedance amplifier. Signal gain is set by U_1 resistance, signal offset is adjusted by the VDAC_1. In addition to the KIT-059 stub, project annotation utilizes few active components (OpAmp2, DAC, IDAC) to visualize PSoC5 on-board hardware, and some external components (SO8, Encoder, NoConnect, TestPoint, Gnd, Power) to complete the circuit.