APPLICATION NOTES

4D SYSTEMS
TURNING TECHNOLOGY INTO ART

ViSi GPIO Bus Control Using DIP Switch Objects Picaso

DOCUMENT DATE: **13th APRIL 2019**
DOCUMENT REVISION: **1.1**

Description

This application note shows how to control the GPIO bus pins using the DIP switch, one of the widgets available in Workshop. The following are needed for this application note:

- Any of the following 4D Picaso touch display modules:

gen4-uLCD-24PT	gen4-uLCD-28PT	gen4-uLCD-32PT
uLCD-24PTU	uLCD-32PTU	uVGA-III

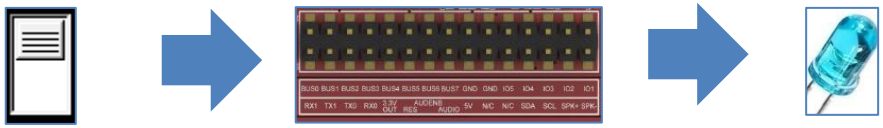
and other superseded modules which support the ViSi Genie environment

- 4D Programming Cable / uUSB-PA5/uUSB-PA5-II for non-gen4 displays(uLCD-xxx)
- 4D Programming Cable & gen4-PA, / gen4-IB / 4D-UPA for gen4 displays (gen4-uLCD-xxx)
- micro-SD (µSD) memory card
- Workshop 4 IDE (installed according to the installation document)
- A breadboard, eight pieces of 1kΩ resistors, and eight pieces of LEDs.
- When downloading an application note, a list of recommended application notes is shown. It is assumed that the user has read or has a working knowledge of the topics presented in these recommended application notes.

Content

Application Overview

Many of the widgets can be used as a virtual input or output device - to send or receive signals to and from external devices through the thirteen available GPIO pins of the Picaso processor. This application note specifically uses the DIP switch to control eight LEDs on a breadboard through the 8-bit GPIO bus. A simple program for controlling the brightness of an LED is also included. In more advanced applications, the LEDs can be replaced with a microcontroller.



DIP switch GPIO 8-bit Bus LEDs

Setup Procedure

For instructions on how to launch Workshop 4, how to open a **ViSi** project, and how to change the target display, kindly refer to the section "**Setup Procedure**" of the application note

ViSi Getting Started - First Project for Picaso and Diablo16

Create a New Project

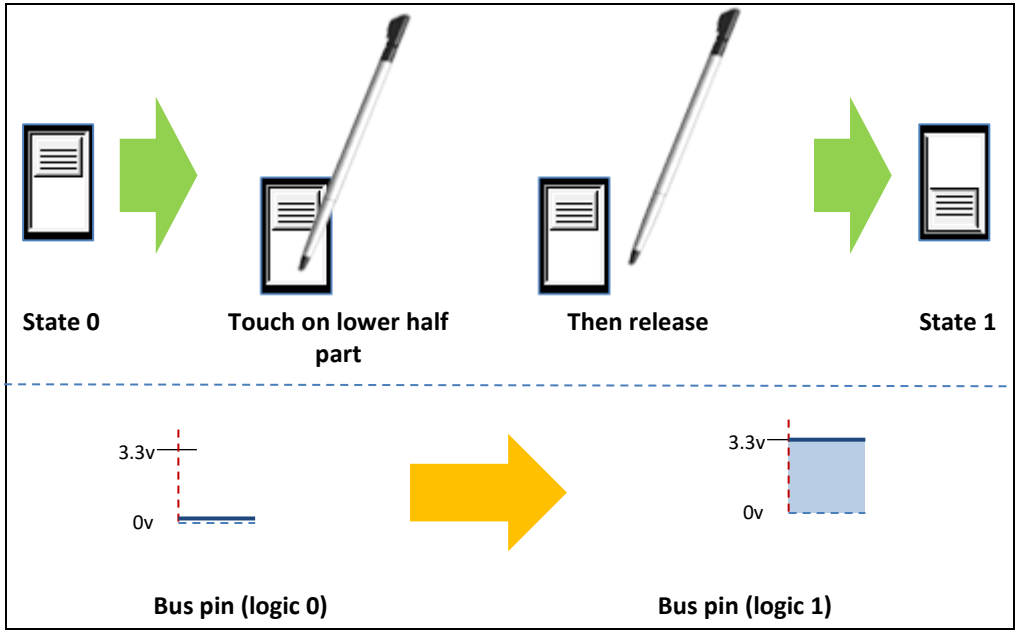
For instructions on how to create a new **ViSi** project, please refer to the section "**Create a New Project**" of the application note

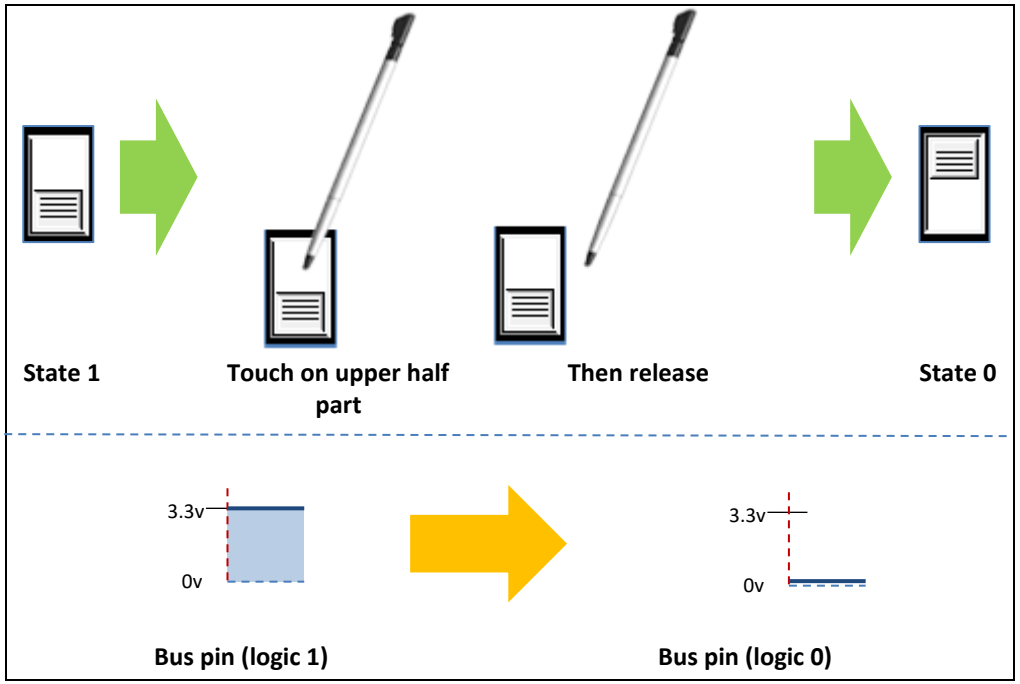
ViSi Getting Started - First Project for Picaso and Diablo16

Design the Project

Define a Working Model

The process of adding a DIP switch and configuring it to respond to touch is discussed in the application note ViSi DIP Switch. The code developed in the said application note will be used here. Only this time, eight DIP switch objects are added, each of which is configured to control a GPIO bus pin. To illustrate:





In 4DGL, the model above can be coded as follows:

```
bus_Set(0x00); //set all GPIO bus pins as outputs
bus_Out(0x00); //then set them to low

repeat
state := touch_Get(TOUCH_STATUS);
if(state == TOUCH_RELEASED)
   n := img_Touched(hndl,iDipswitch1);
   if(n == iDipswitch1)
       x := touch_Get(TOUCH_GETX);
       y := touch_Get(TOUCH_GETY);
       DIPstate :=  (y - 100) / (60/2);
       img_SetWord(hndl, iDipswitch1, IMAGE_INDEX, DIPstate);
       img_Show(hndl, iDipswitch1);
```

```
      //control BUS_0
      if(DIPstate == 0)
         pin_LO(BUS_0);
      else
         pin_HI(BUS_0);
      endif
   endif
endif
forever
```

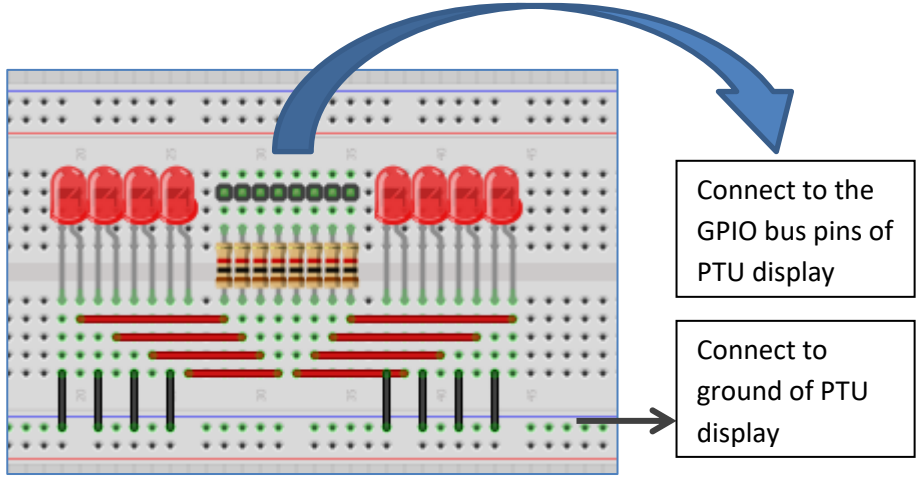
The code above is for a single DIP switch that turns BUS_0 on and off. To control the entire GPIO bus, create and configure eight DIP switch objects in the WYSIWYG screen. Insert the generated code for the eight DIP switch objects to the main code. This application note comes with four ViSi files, one of which is for controlling the GPIO bus using eight DIP switch objects.

Create a Test Circuit

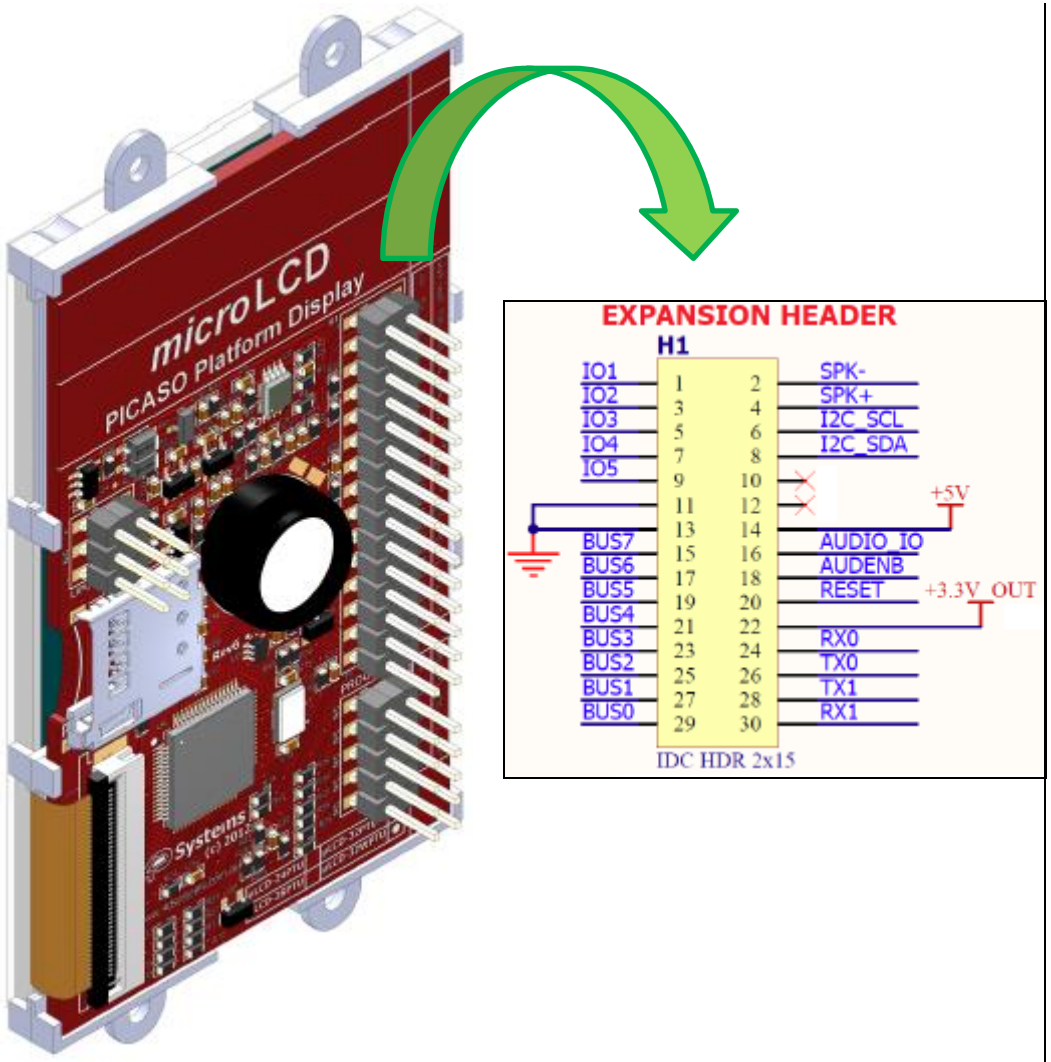
The user can build a simple LED circuit for testing I/O related programs. Eight LEDs, each in series with a **1 kΩ resistor**, are connected to the GPIO bus pins to indicate their status.

Schematic:

Breadboard Layout:



*Schematic and breadboard layout diagrams were made using **Fritzing**



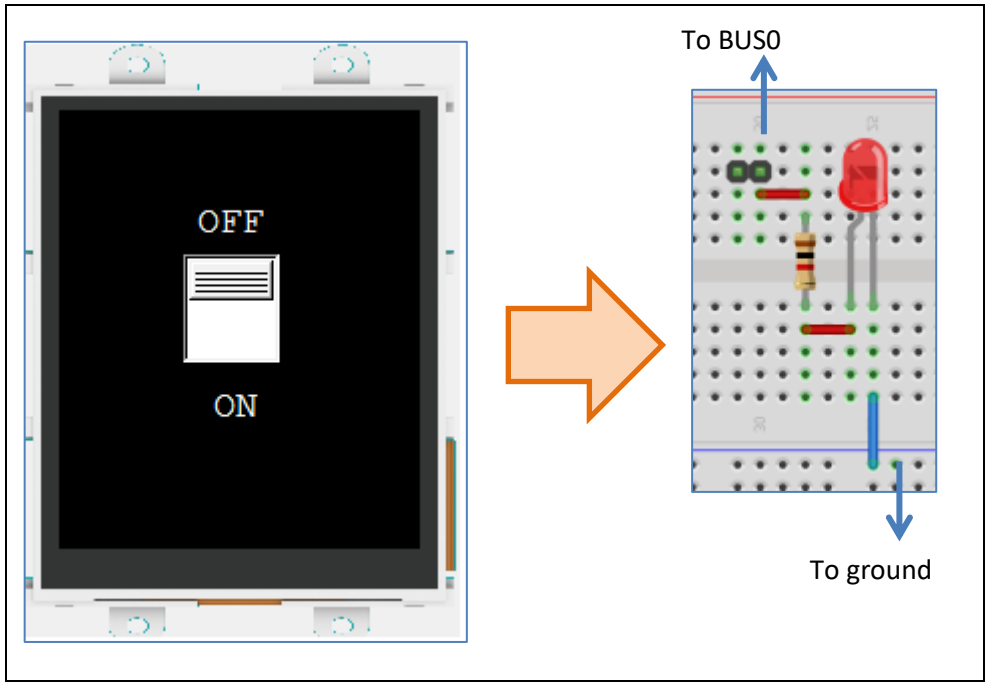
Shown above is the backside of a uLCD-32WPTU and the location of the bus pins (BUS0 to BUS7). Consult the datasheet of your display module for the pin configuration and board layout.

Attached ViSi Files

Four ViSi files are attached for you to experiment with. In each file, many of the lines are commented to help explain the flow of the program.

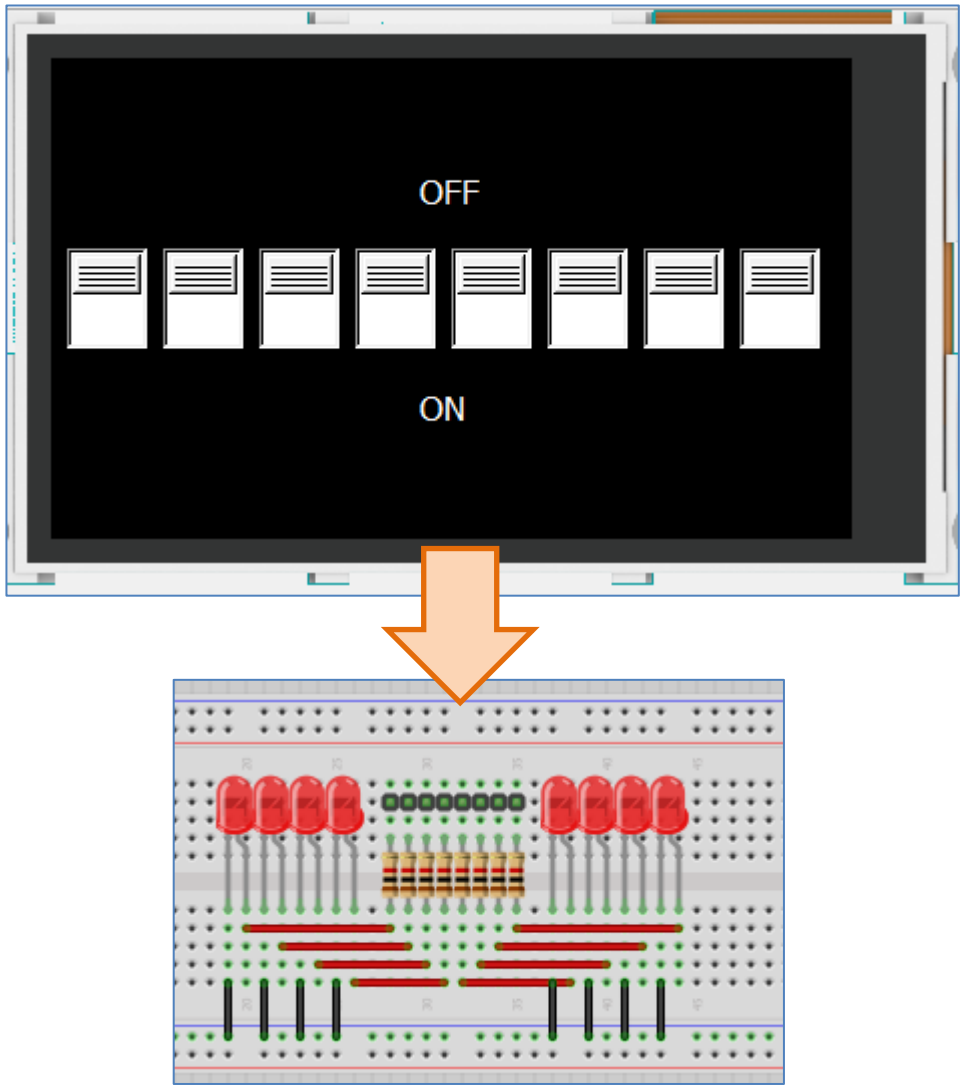
DIPSwitchTutorialBUS_0.4dViSi

A DIP switch object controls the BUS_0 pin.



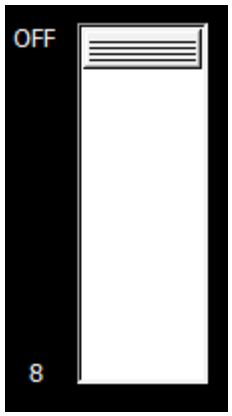
GPIOBusControl_8DIPSwitchObj.4dViSi

In this program, eight DIP switch objects control the GPIO bus such that when a DIP switch is at state 0 or 1, the corresponding LED is turned off or on, respectively.



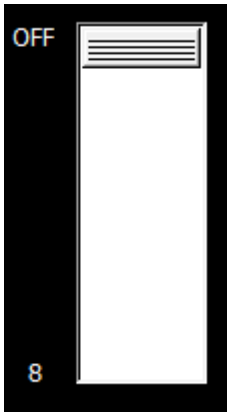
GPIOBusControlUsingDIPSwitch.4dViSi

This program controls the GPIO bus using a single DIP switch with nine positions. When the DIP switch is at state 0, the eight LEDs are turned off. When the switch is at state 1, LED1 is turned on, and the rest are turned off. When the switch is at state 2, LED2 is turned on, and the rest are turned off. The same is the case with states 3 to 8.



DIP switch state	LED state
0	
1	
2	
3	
4	
5	
6	
7	
8	

The program can also be modified such that the LEDs will display the DIP switch state in binary. Only four LEDs are needed in this case.

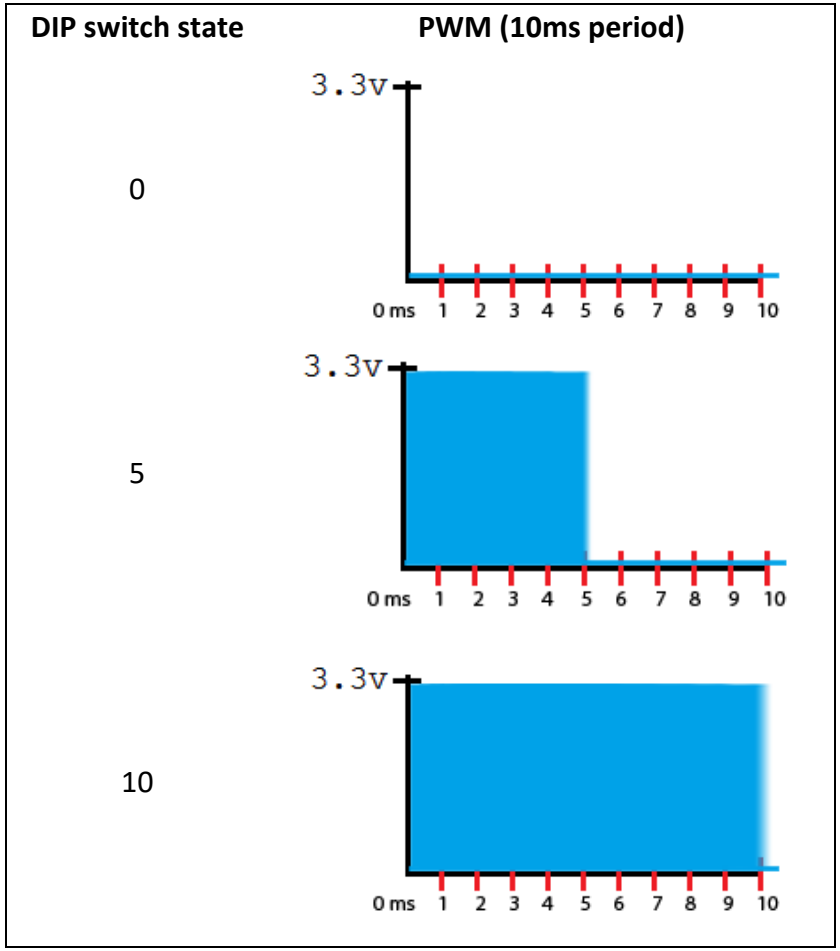


DIP switch state	LED state
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0

Using the 8 LEDs, 2^8 = 256 states can be displayed in binary. Aside from the DIP switch, a multistate knob or rotary switch can be used as an input for this purpose. Use the function **bus_Out**(arg1) instead of **pin_HI**(pin).

GPIOBusControlUsingDIPSwitchPWM.4dViSi

In this program, the brightness of an LED connected to BUS_0 and in series with a 1kΩ resistor is controlled using a DIP switch with 11 positions.



Run the Program

For instructions on how to save a **ViSi** project, how to connect the target display to the PC, how to select the program destination (this option is not available for Goldelox displays), and how to compile and download a program, please refer to the section "**Run the Program**" of the application note

ViSi Getting Started - First Project for Picaso and Diablo16

The uLCD-32PTU and/or uLCD-35DT display modules are commonly used as examples, but the procedure is the same for other displays.

Proprietary Information

Disclaimer of Warranties & Limitation of Liability